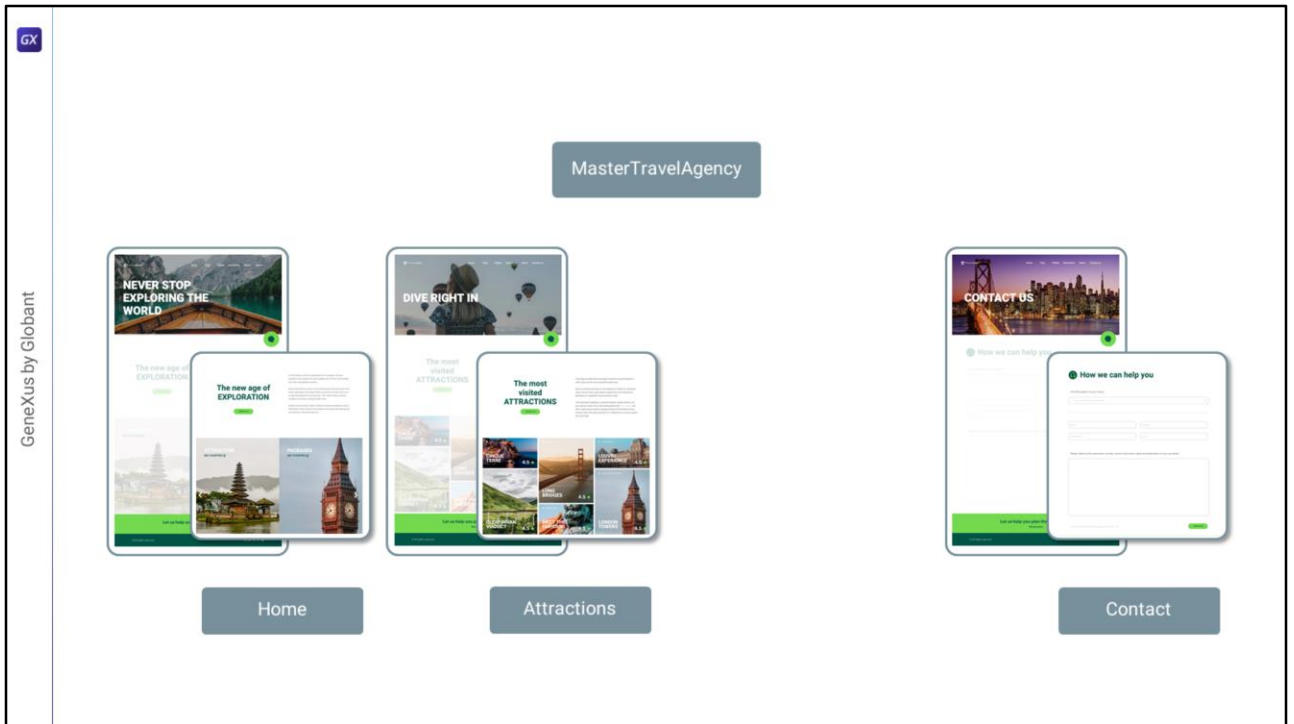


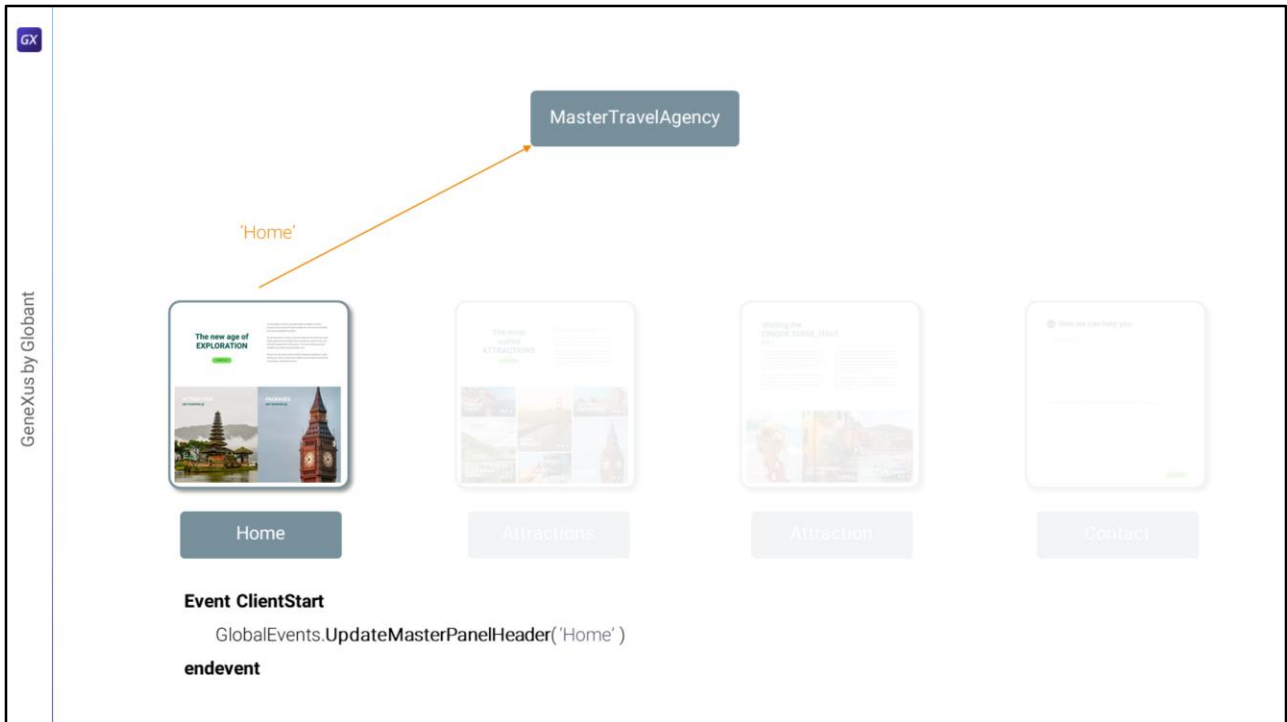
Master Panel: Header update and Navigations (part 2)



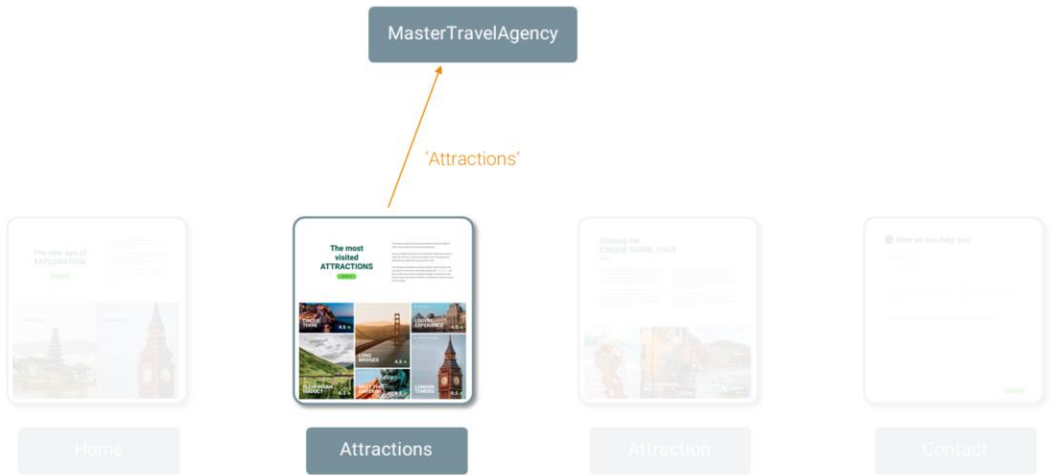
Cecilia Fernández



No vídeo anterior implementamos as navegações e a carga correta do Header do Master Panel...



...de acordo com, se estava carregando o Home...



Event ClientStart

```
GlobalEvents.UpdateMasterPanelHeader('Attractions')
```

endevent

...Attractions...

MasterTravelAgency

'Contact'



Home



Attractions



Attraction



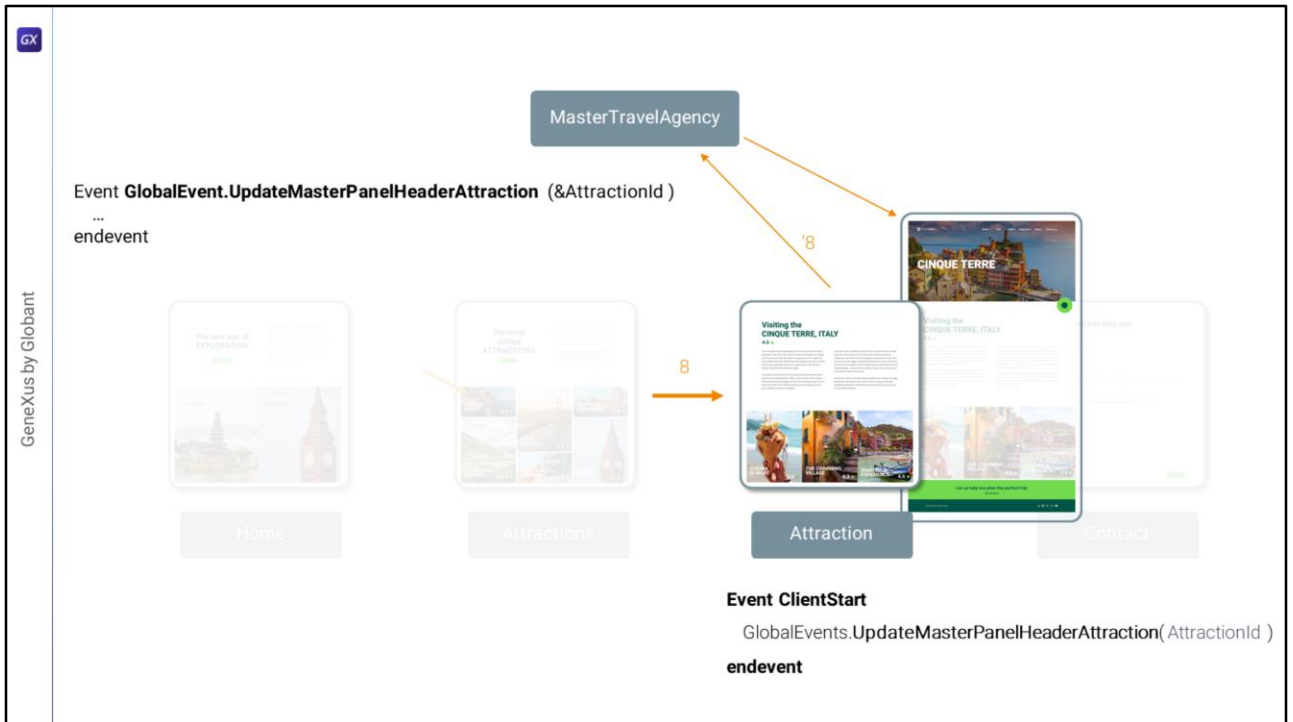
Contact

Event ClientStart

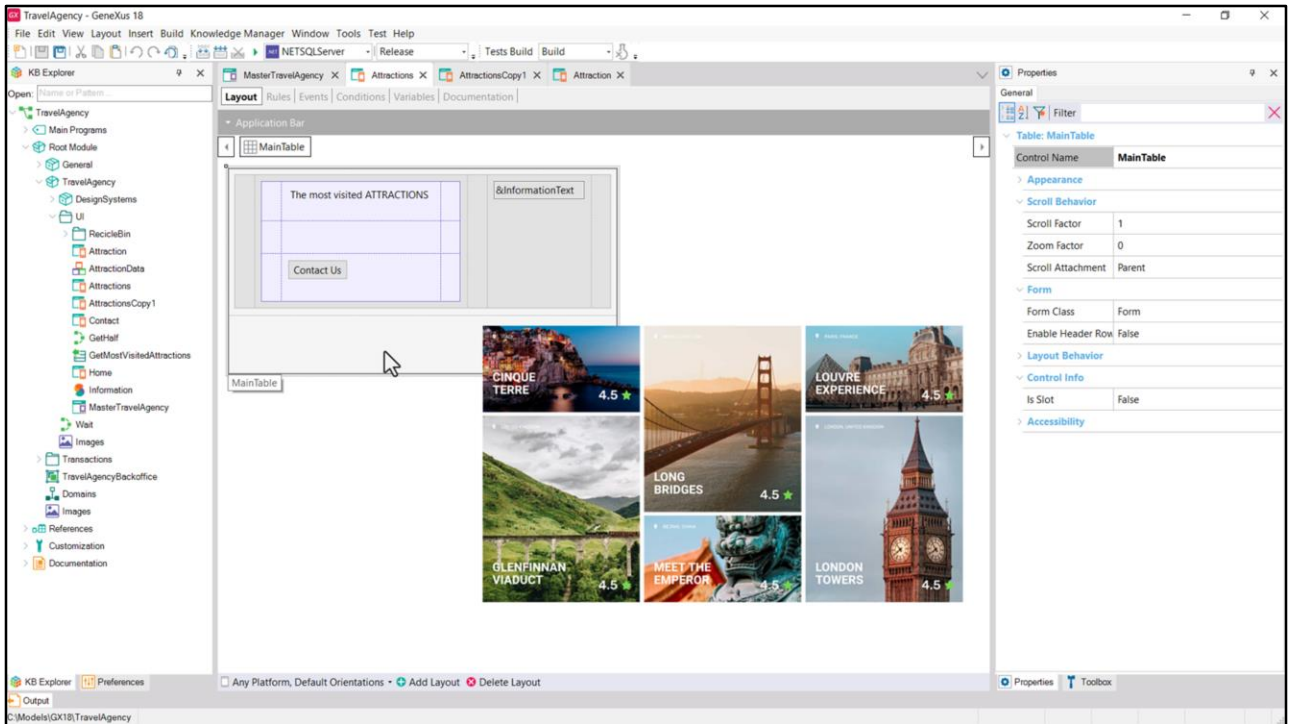
```
GlobalEvents.UpdateMasterPanelHeader('Contact')
```

endevent

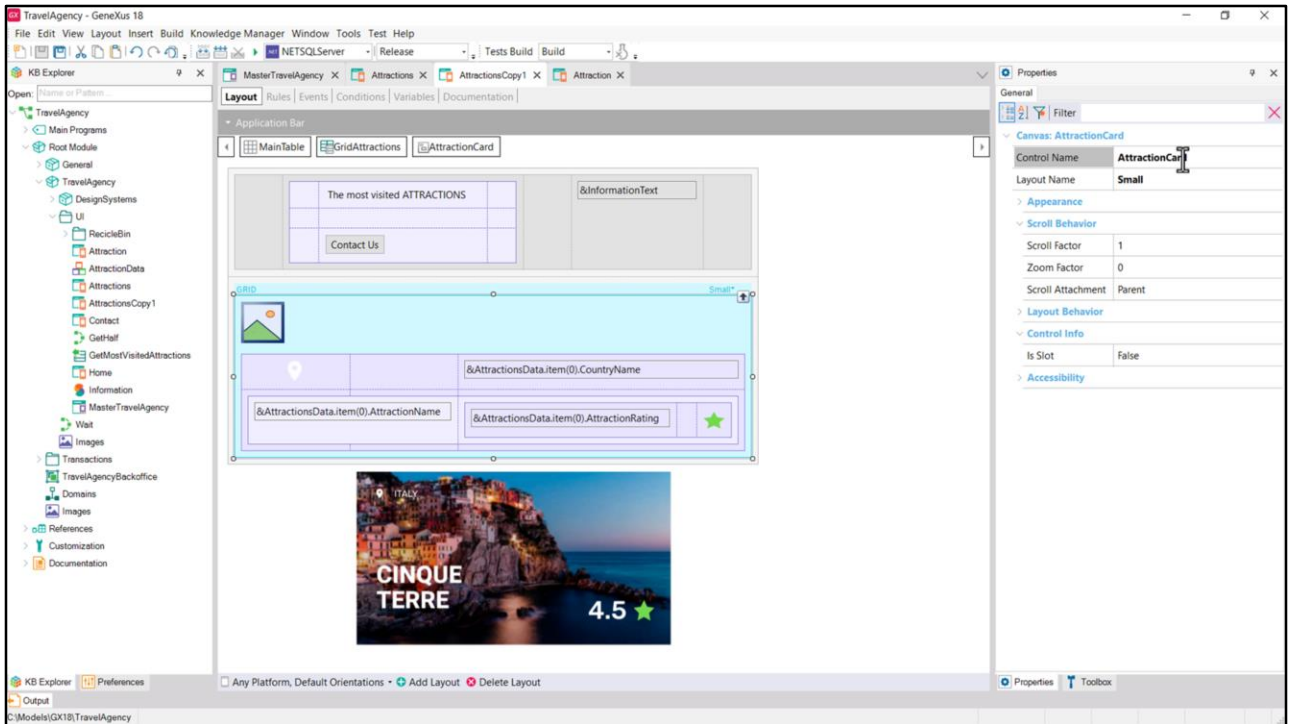
...ou Contact...



Ficou pendente implementar que o Master Panel carregue no Header a imagem e título da atração que recebeu por parâmetro, o panel Attraction, ao ser executado. E que o recebeu quando o usuário selecionou uma atração turística neste grid.

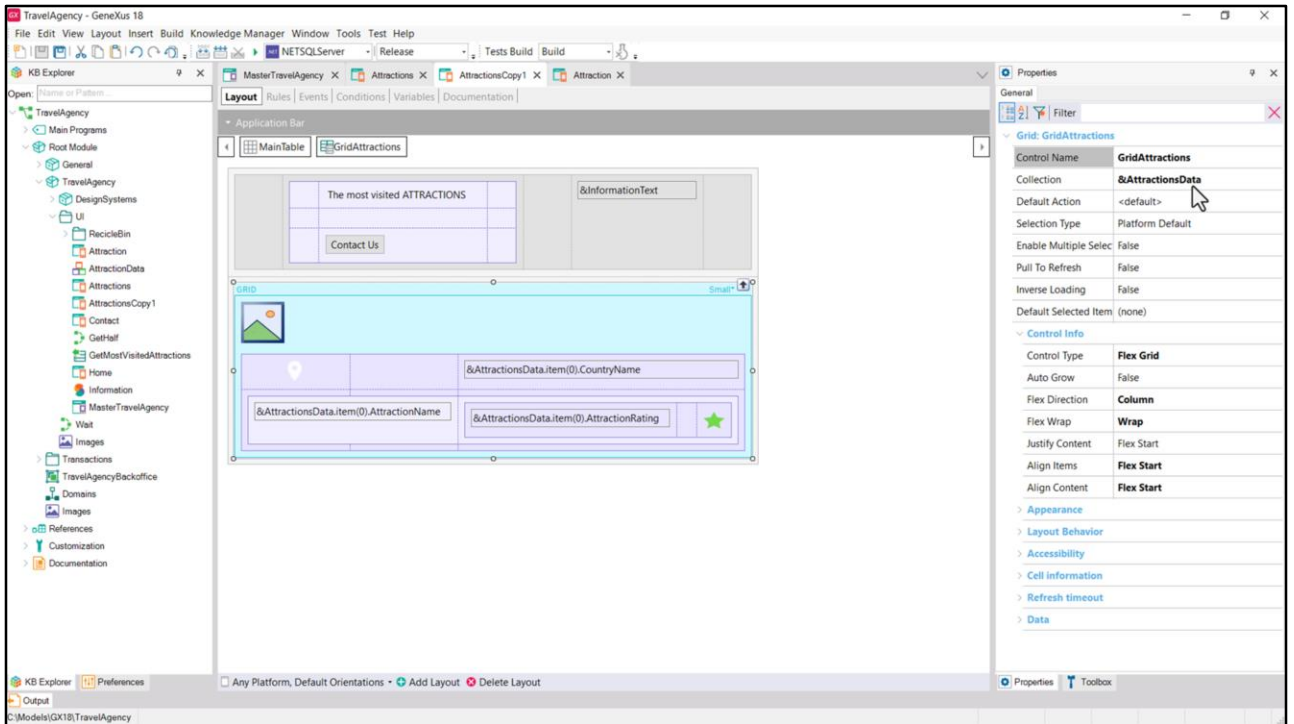


Para poder implementar isso e testá-lo, então, precisarei primeiro adicionar ao panel Attractions, nesta segunda linha, o carrossel com as atrações turísticas carregadas da base de dados...

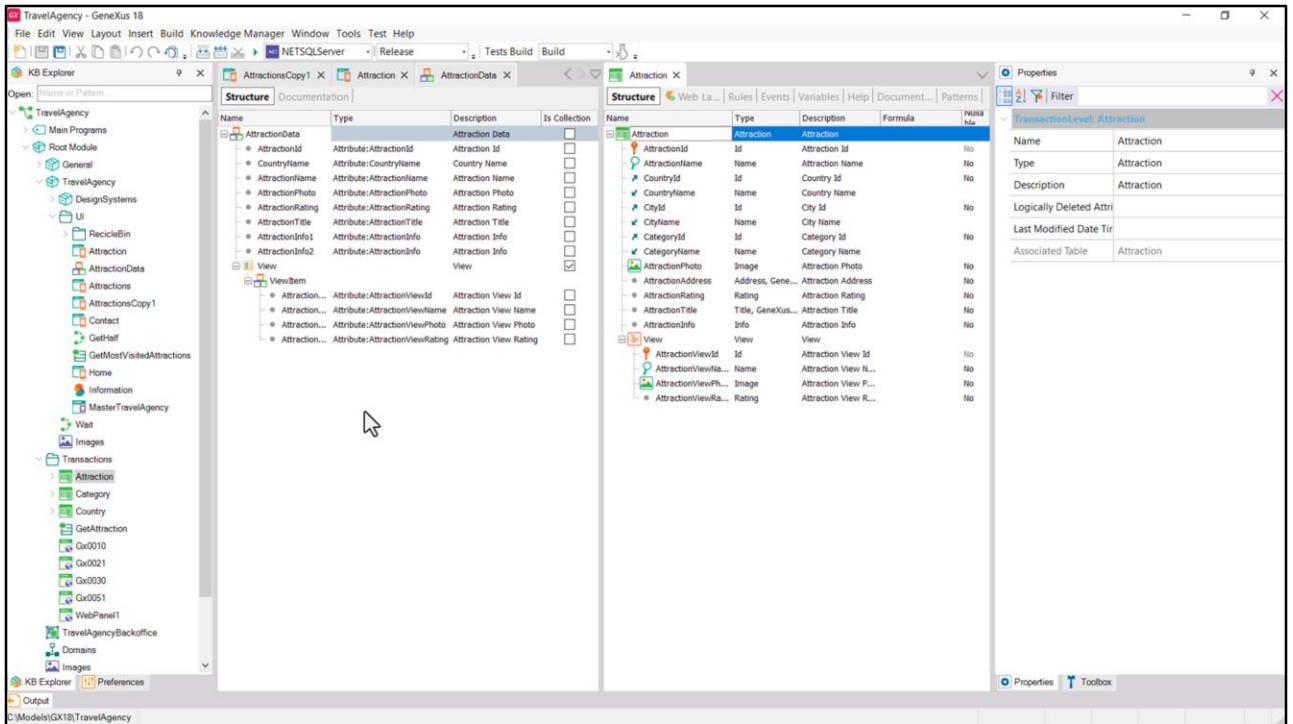


Aqui fiz uma cópia do panel e o implementei para que não percamos tempo agora fazendo isso (em outro vídeo veremos essa implementação em detalhe, não nos preocupemos neste momento). Apenas reparamos que se trata de um grid, por isso carrega dinamicamente seus itens.

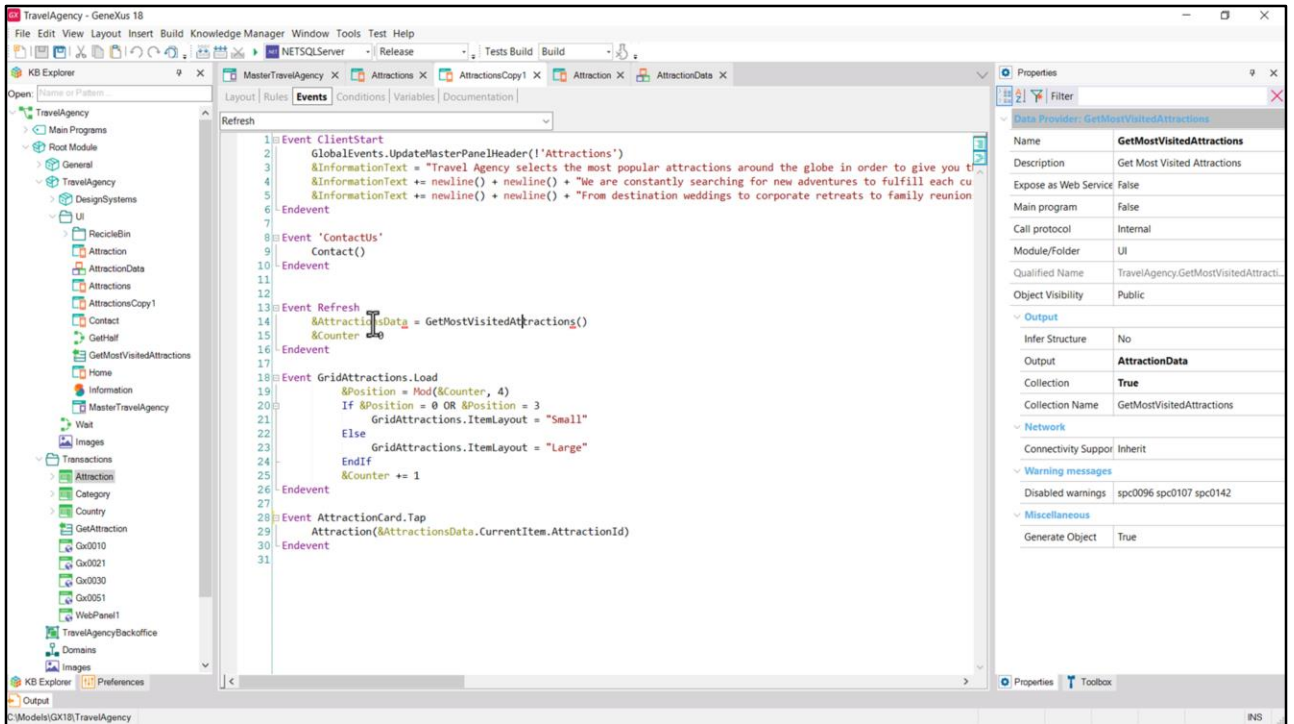
Cada item se trata de um canvas que chamei de `AttractionCard`, justamente para renderizar cada uma das atrações turísticas como um card.



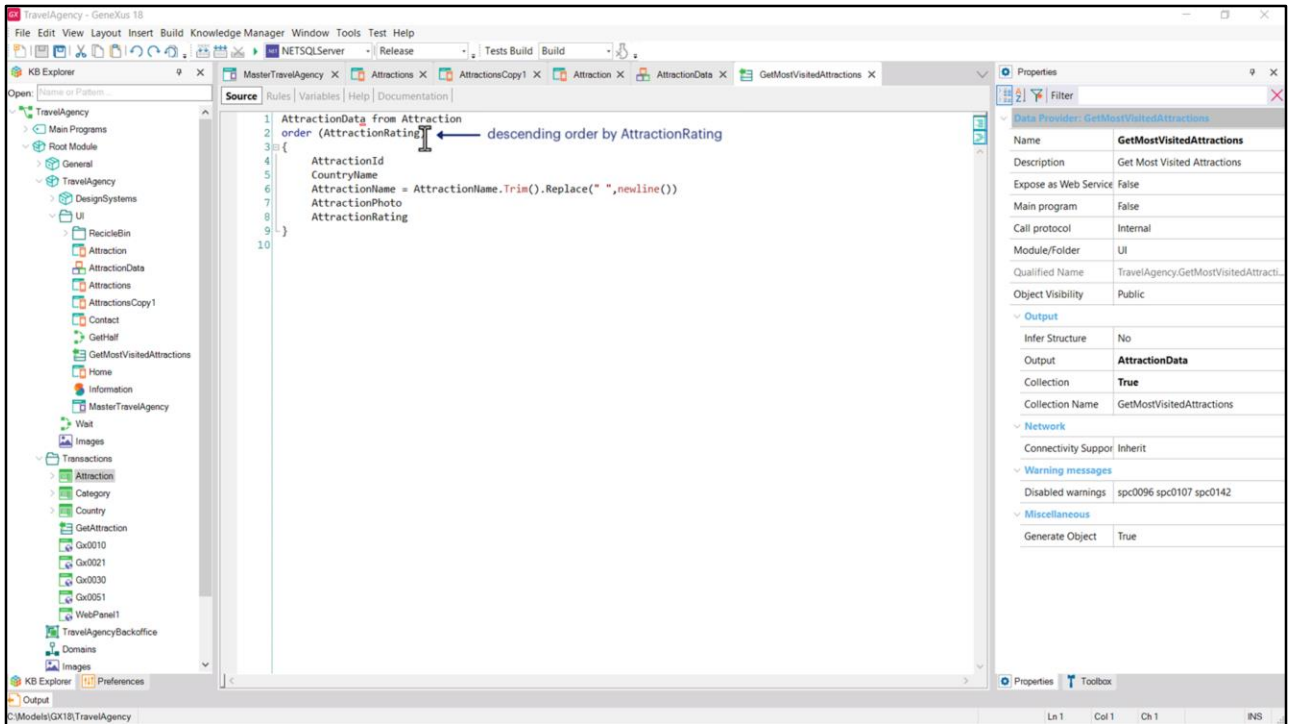
Na verdade, existem muitas maneiras possíveis de implementar este grid, aqui escolhi uma, onde carrego os itens do grid nesta variável que é do tipo de dados uma coleção deste SDT.



Vejam que é quase idêntico à estrutura da transação Attraction. Eu o criei com os campos que vou precisar para os dois painéis Attractions e Attraction.

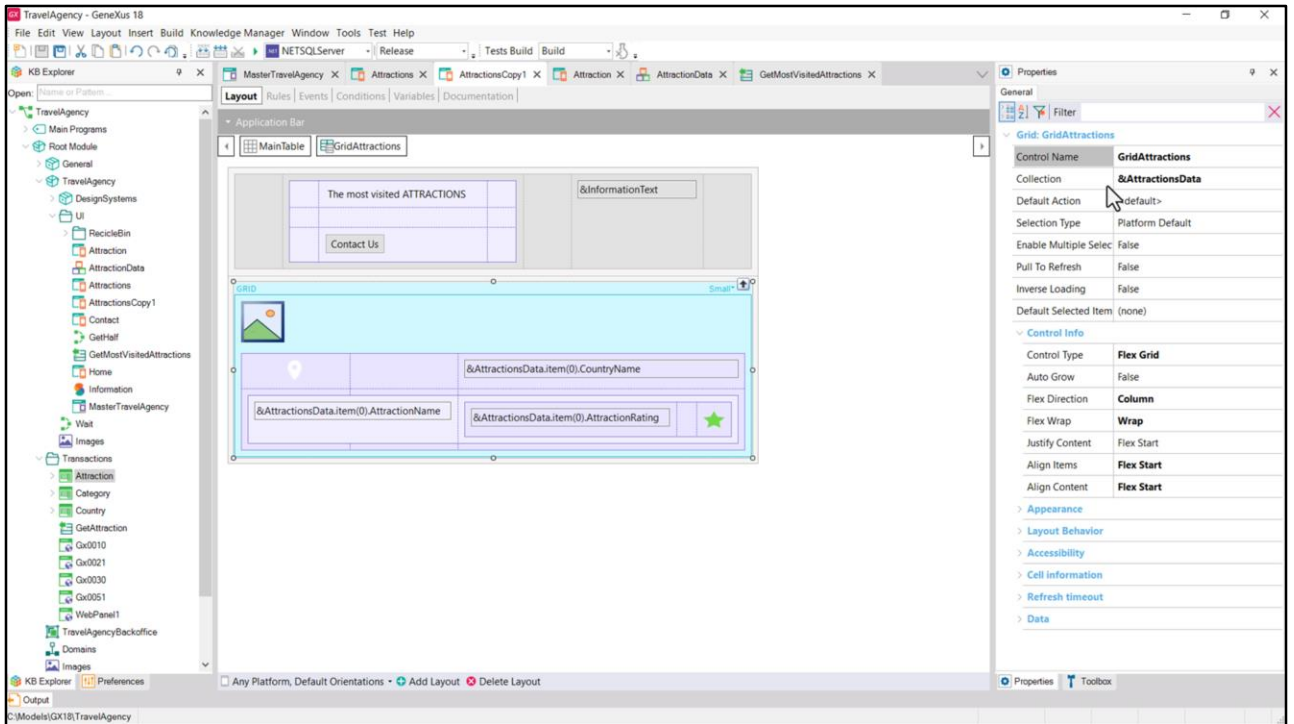


Em resumo, o que faço aqui é chamar este data provider...

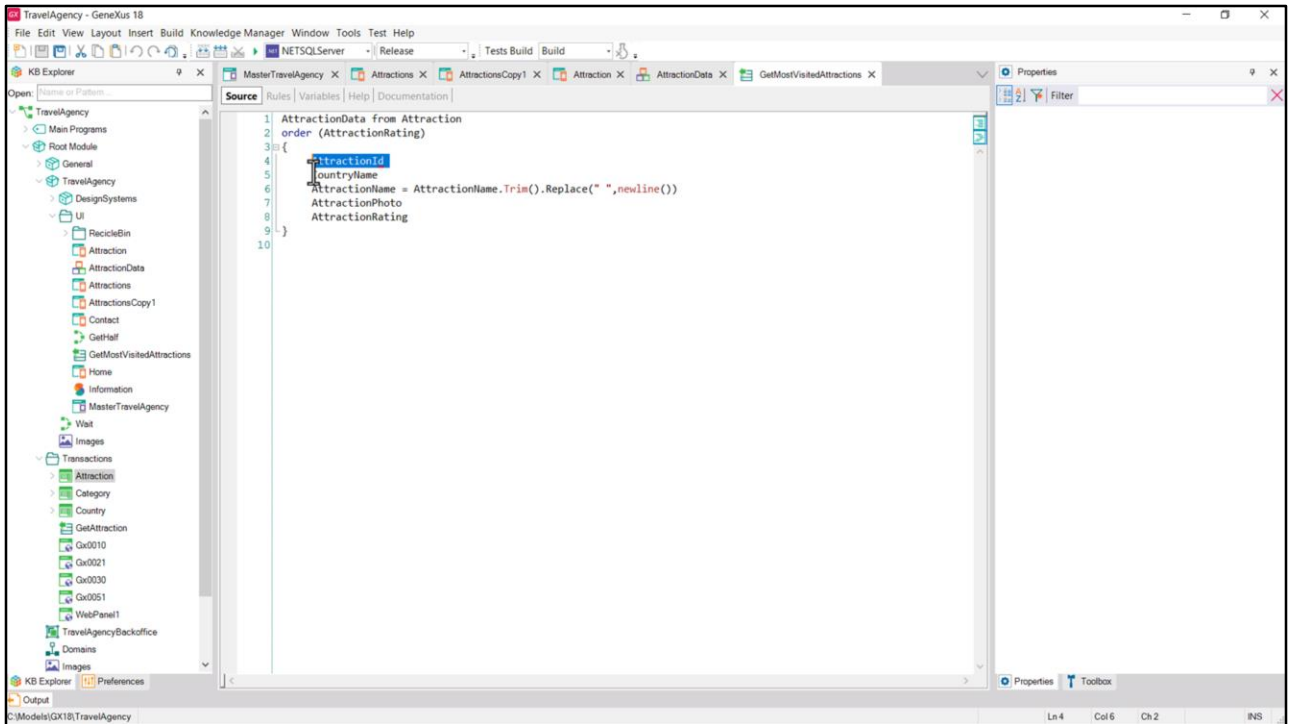


...que devolverá a coleção do SDT carregada a partir da base de dados.

Aqui estamos indicando os campos de cada item que nos interessa obter para mostrar em cada Card. Vejam que não estamos carregando todos os elementos, mas apenas os que nos interessam nesta oportunidade.

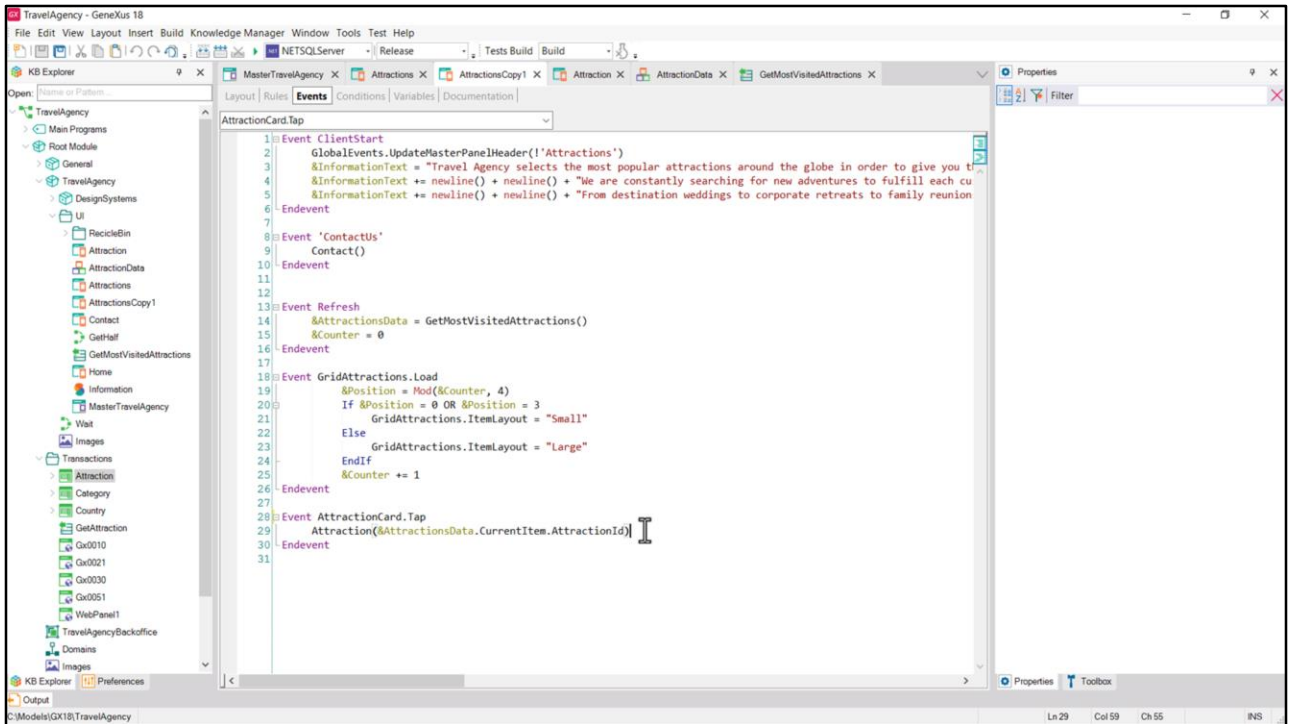


Esta coleção será a que será carregada automaticamente no grid, pois assim indiquei em sua propriedade Collection.

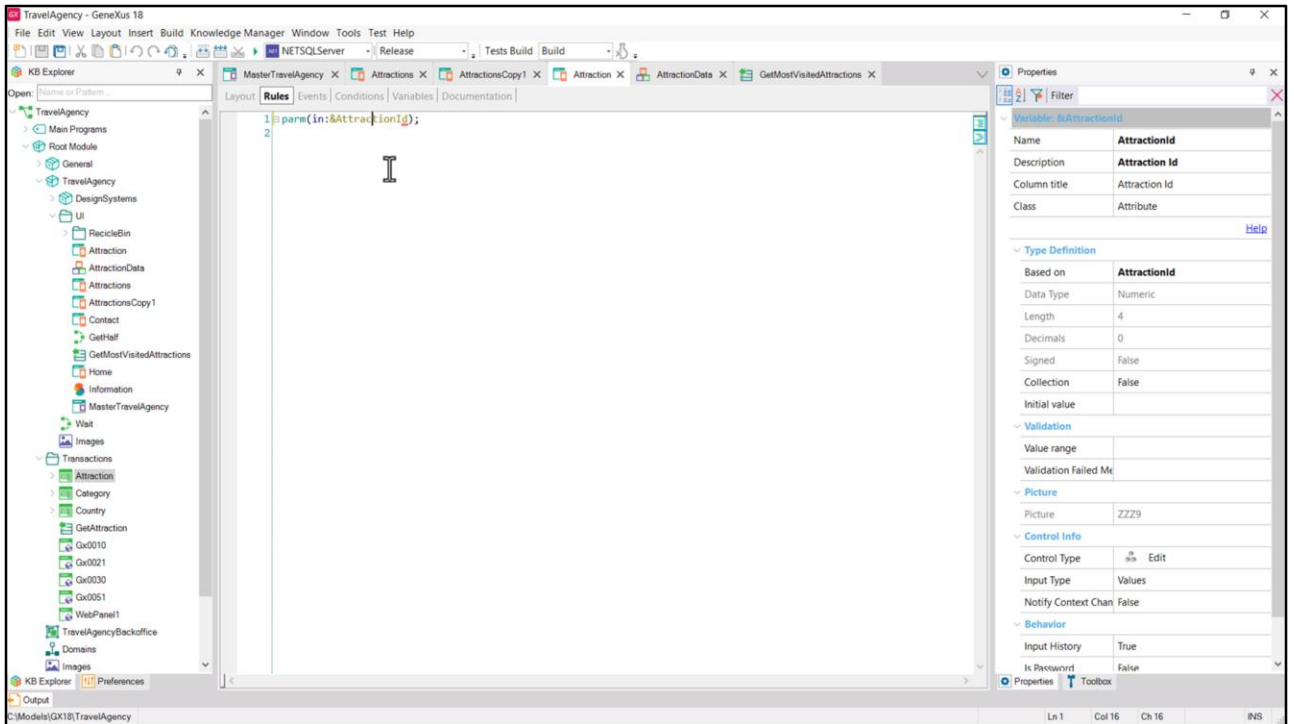


Repito que não importa agora entender isso. Basta saber que este grid será carregado a partir deste SDT coleção que, por sua vez, estará carregado com a informação que nos interessava da base de dados.

Em particular, vejamos que cada item da coleção virá com o AttractionId carregado.

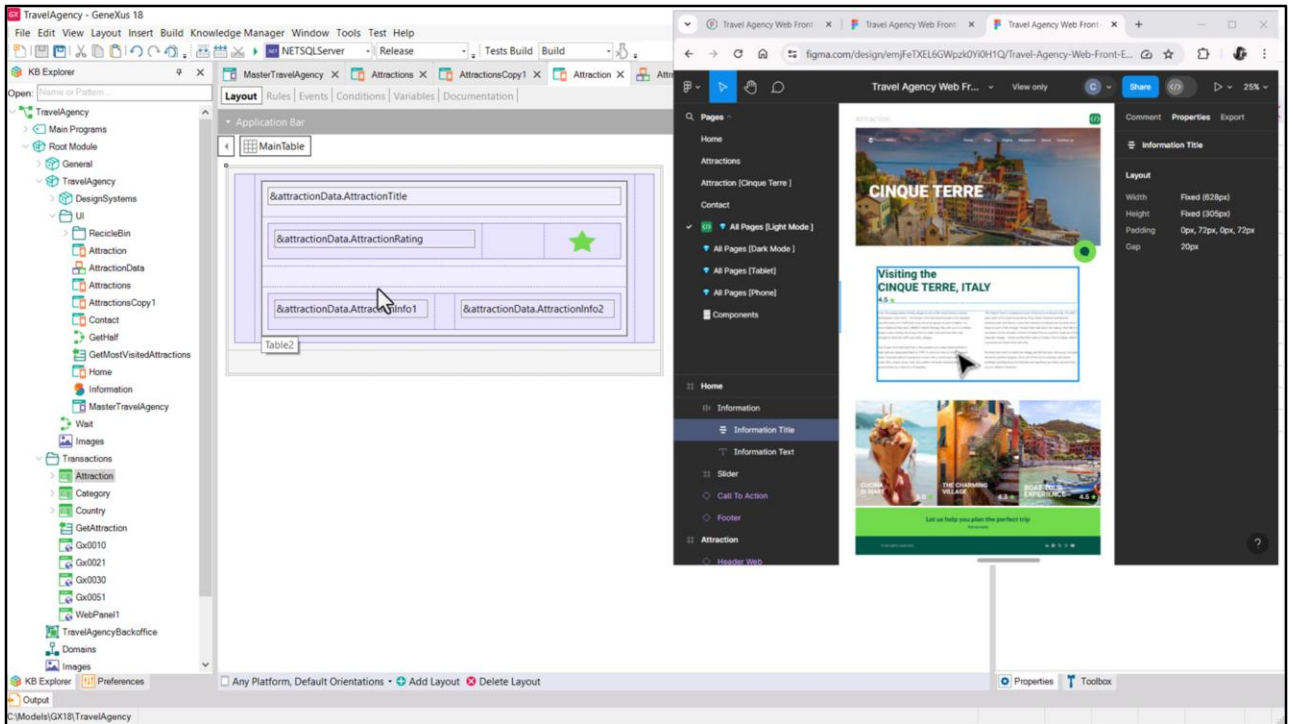


Então, o que fiz aqui foi programar sobre o canvas que corresponde a cada item do grid, o evento tap (que é o mesmo que o click para web). E ali o que estou fazendo é invocar o panel Attraction, passando como parâmetro para o item clicado, o valor de AttractionId.

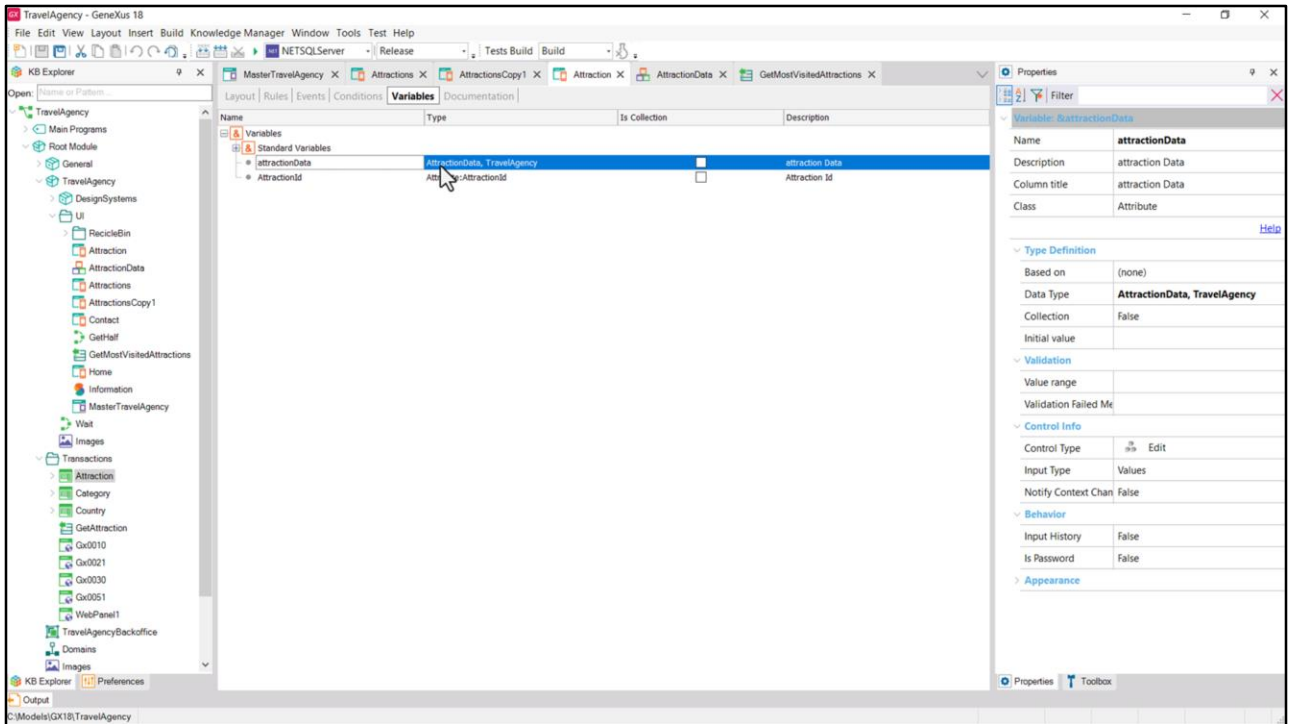


Bem, com isso implementamos a invocação.

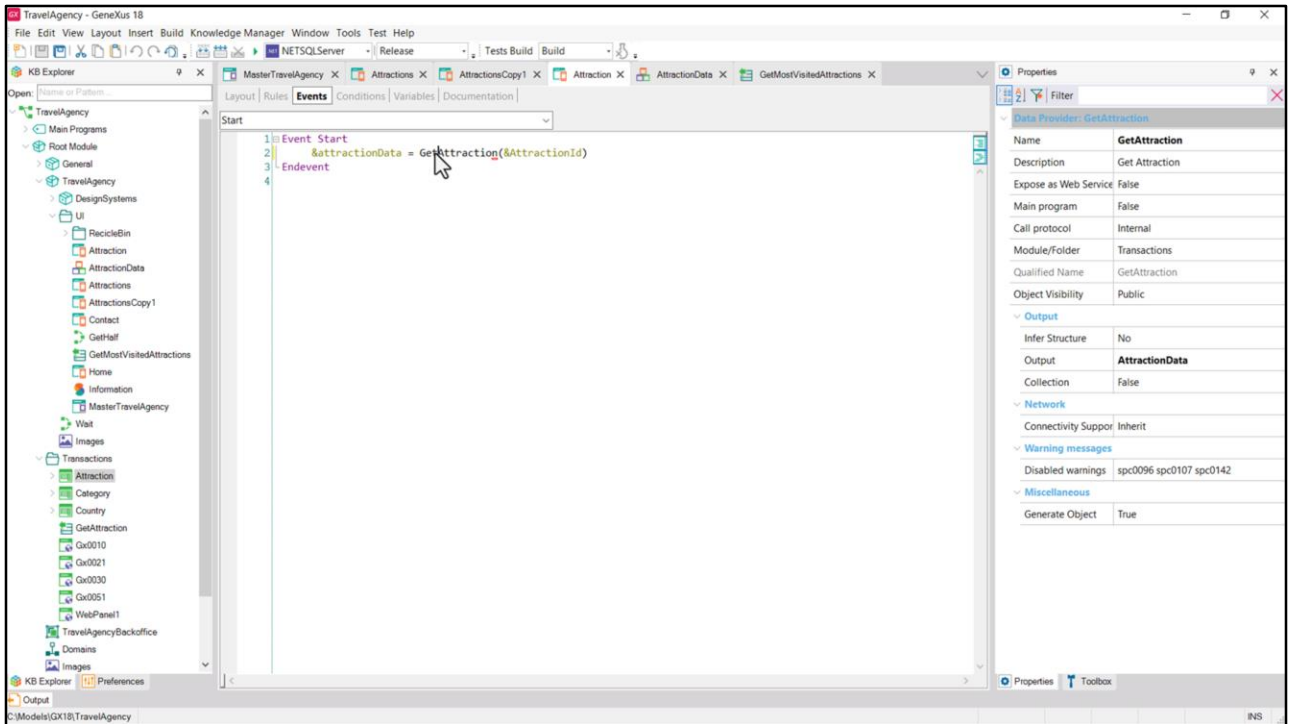
Depois, em Attraction já avancei um pouco também. Vejam que defini a regra parm para que o panel possa receber por parâmetro nesta variável o identificador da atração turística.



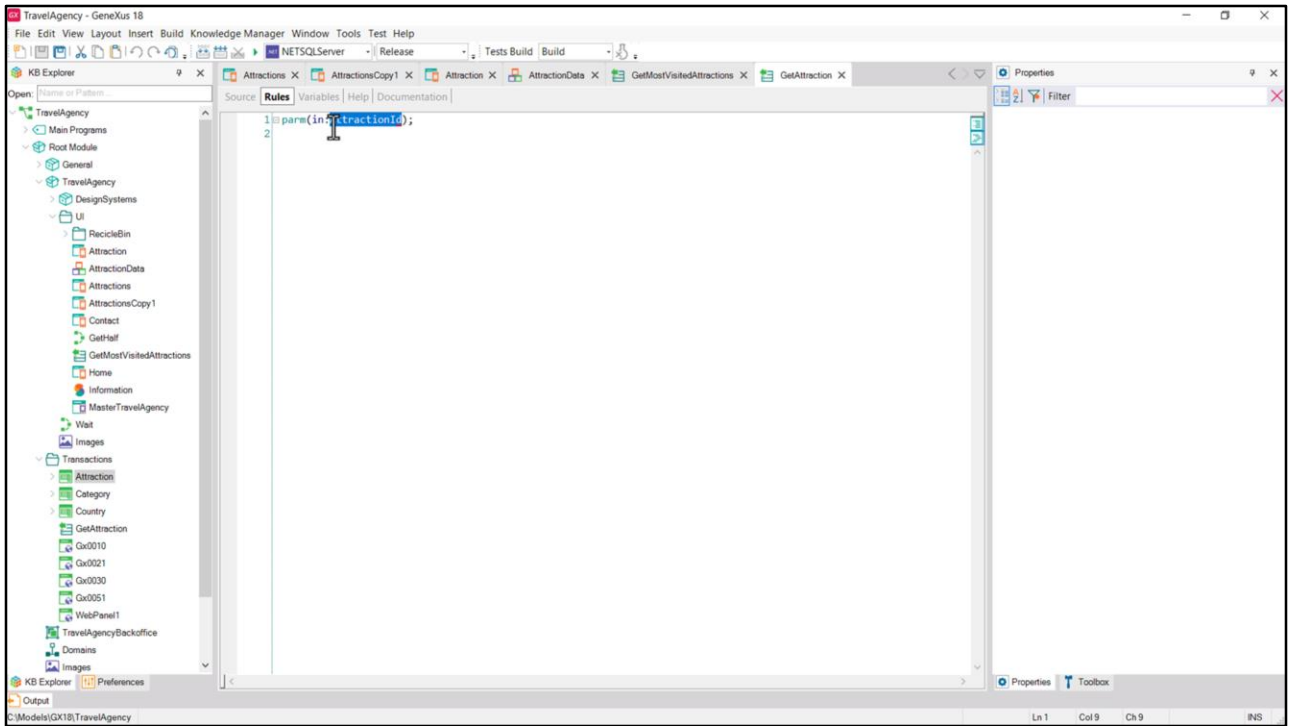
E por enquanto, no seu layout, implementei apenas esta parte...



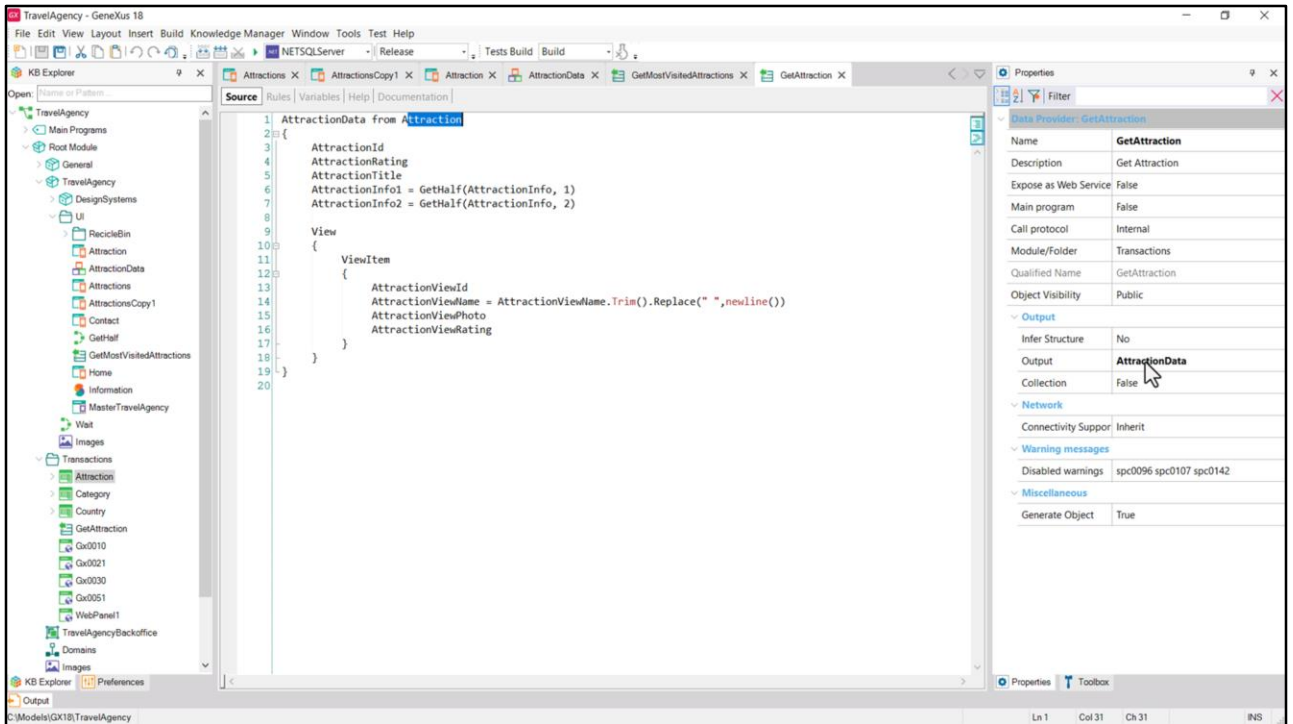
...para a qual utilizo outra variável baseada no mesmo SDT, mas desta vez a variável não é coleção.



E agora chamo outro Data Provider ao qual passo o identificador da atração recebido por parâmetro...



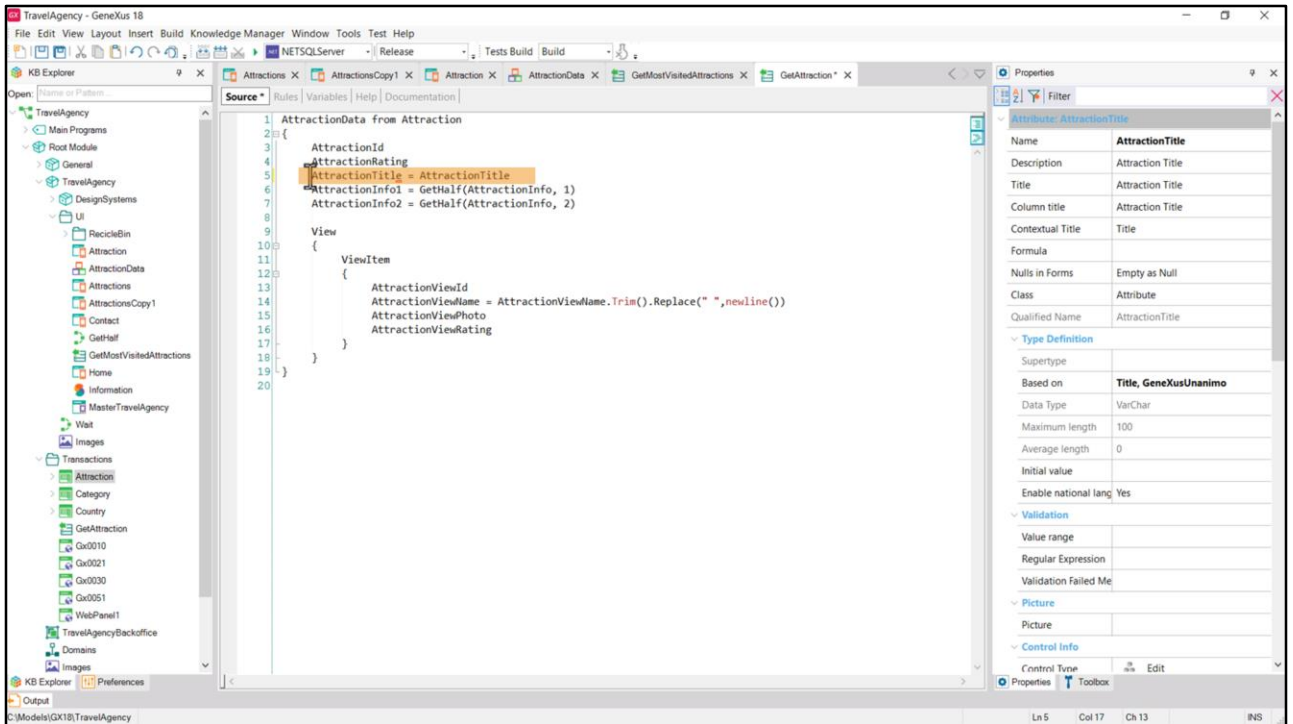
...e este Data Provider recebe esse valor no atributo AttractionId...



...pelo que realizará um filtro automático...ao percorrer neste grupo a transação base Attraction...

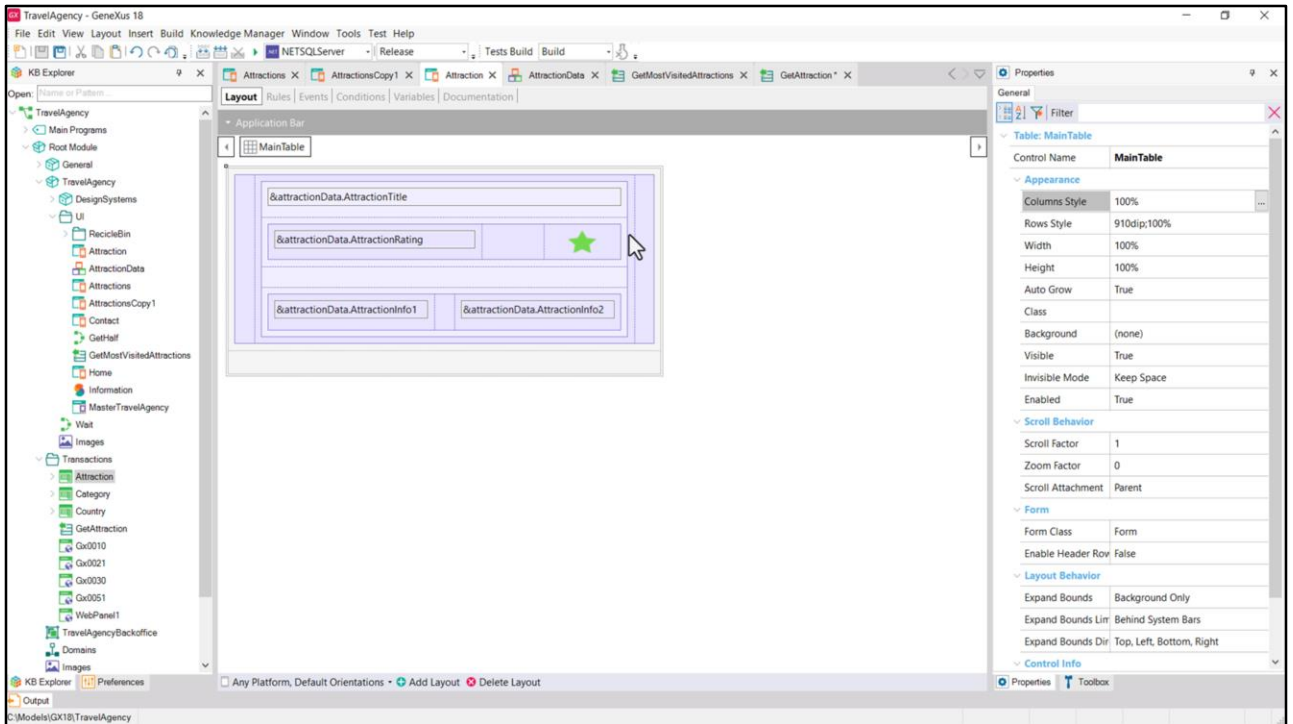
Pelo filtro automático, justamente, ficará somente com o registro da tabela Attraction que possui esse id do parâmetro.

E o que farei será carregar no SDT que será devolvido pelo Data Provider somente o que preciso por enquanto: o id, o rating da atração, o título, a informação dividida em dois blocos de texto (para fazer essa divisão chamo um procedimento que me devolve do atributo de Info a primeira metade se envio 1, a segunda metade se envio 2)...



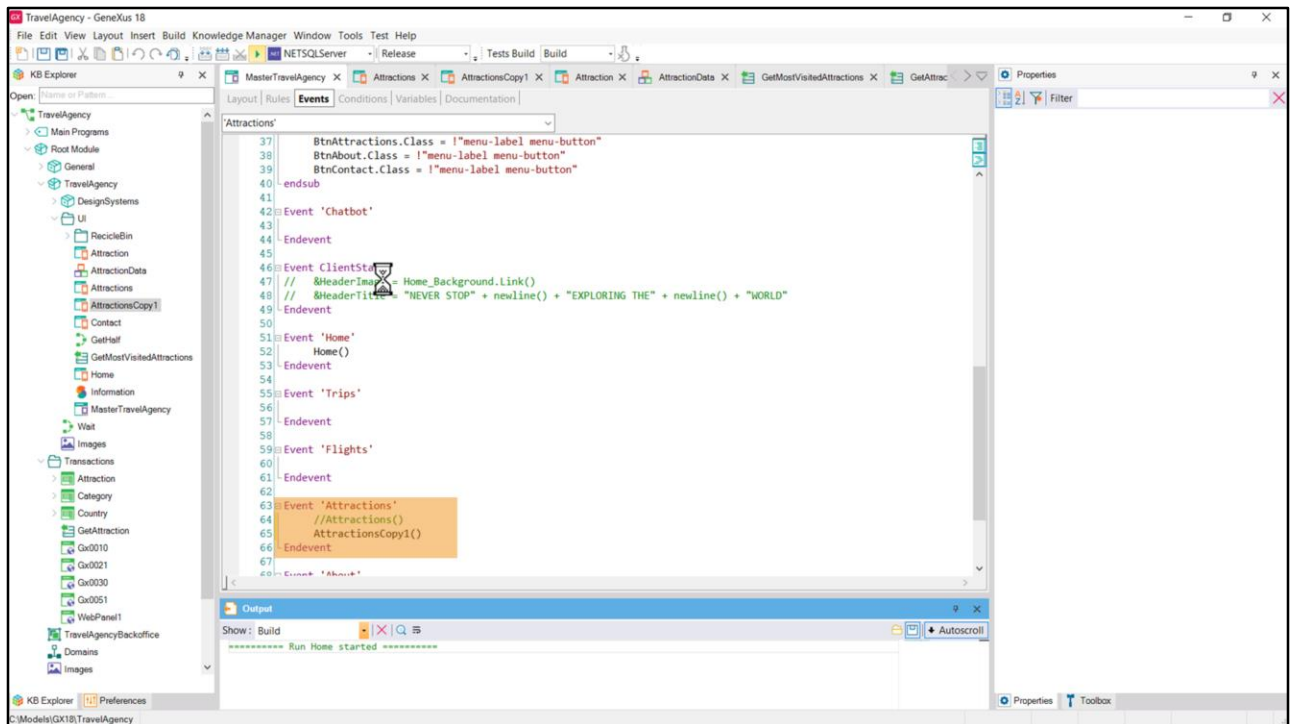
Como lembrete para entender este código: como o elemento do SDT e o atributo da tabela base têm o mesmo nome, então posso usar a notação abreviada, em vez de ter que fazer explicitamente a atribuição. Por exemplo, escrever isto é o mesmo que este outro... Mas, enquanto o da esquerda é o elemento do SDT, o da direita é o atributo da tabela Attraction da base de dados.

O Data Provider também devolverá carregados os itens desta coleção, mas por enquanto utilizaremos apenas seu primeiro nível...

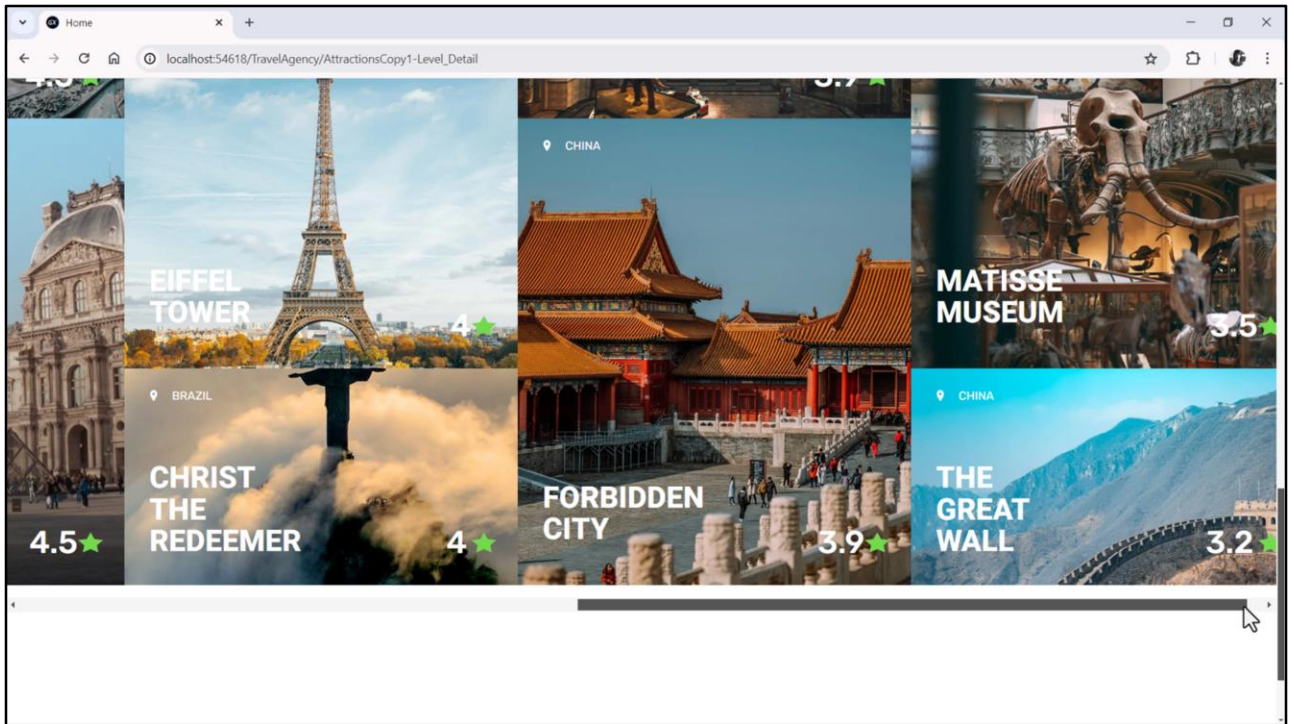


...que é o que utilizamos aqui no layout de Attraction.

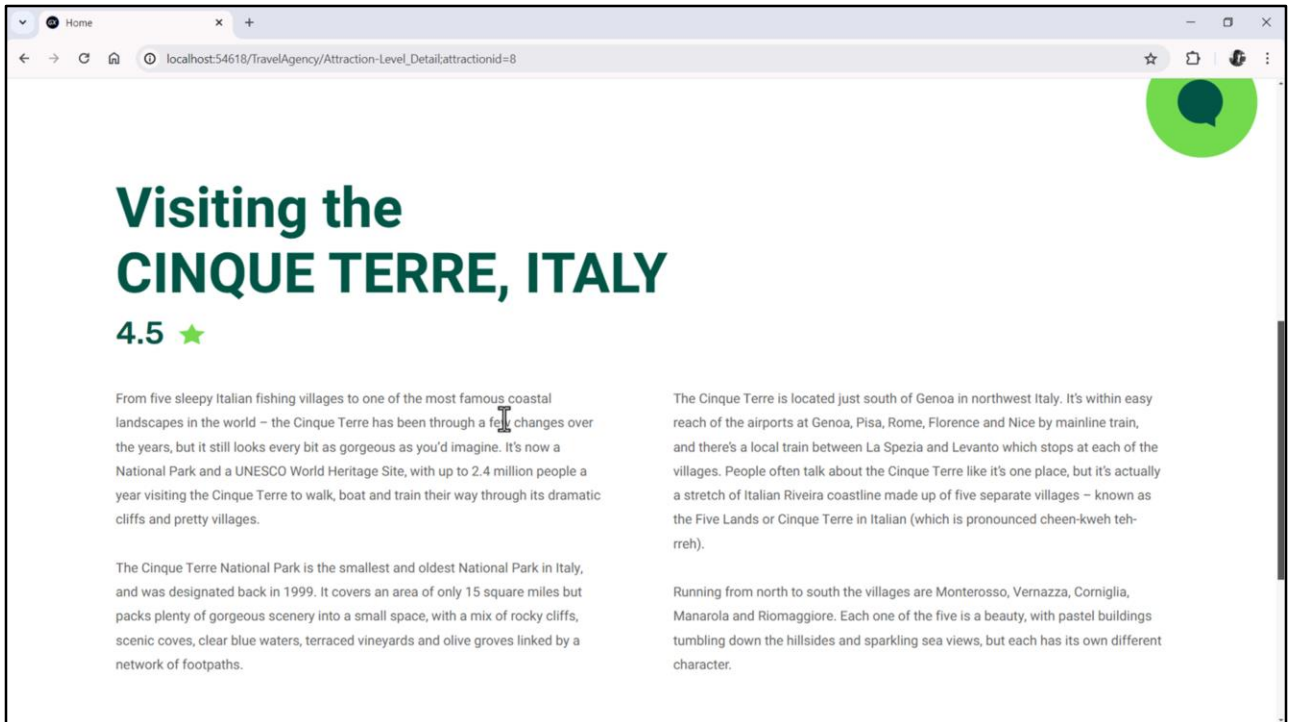
Antes de implementar o que é o objetivo deste vídeo, vou executar isso para que fique bem claro de onde partimos.



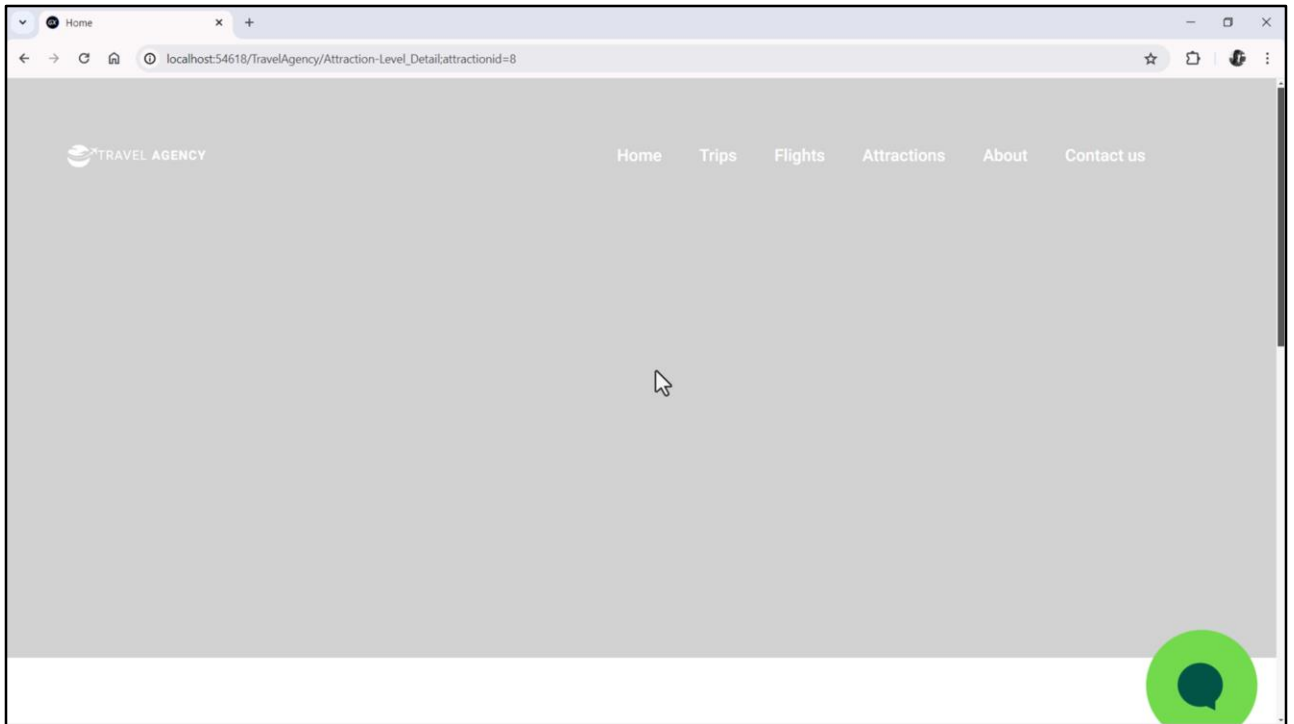
Aqui, no Master Panel, no evento associado ao botão do Menu Attractions, estou chamando o panel Attractions que não possui o grid. Então, o que vou fazer é trocar a invocação pelo panel que o possui. Executo...



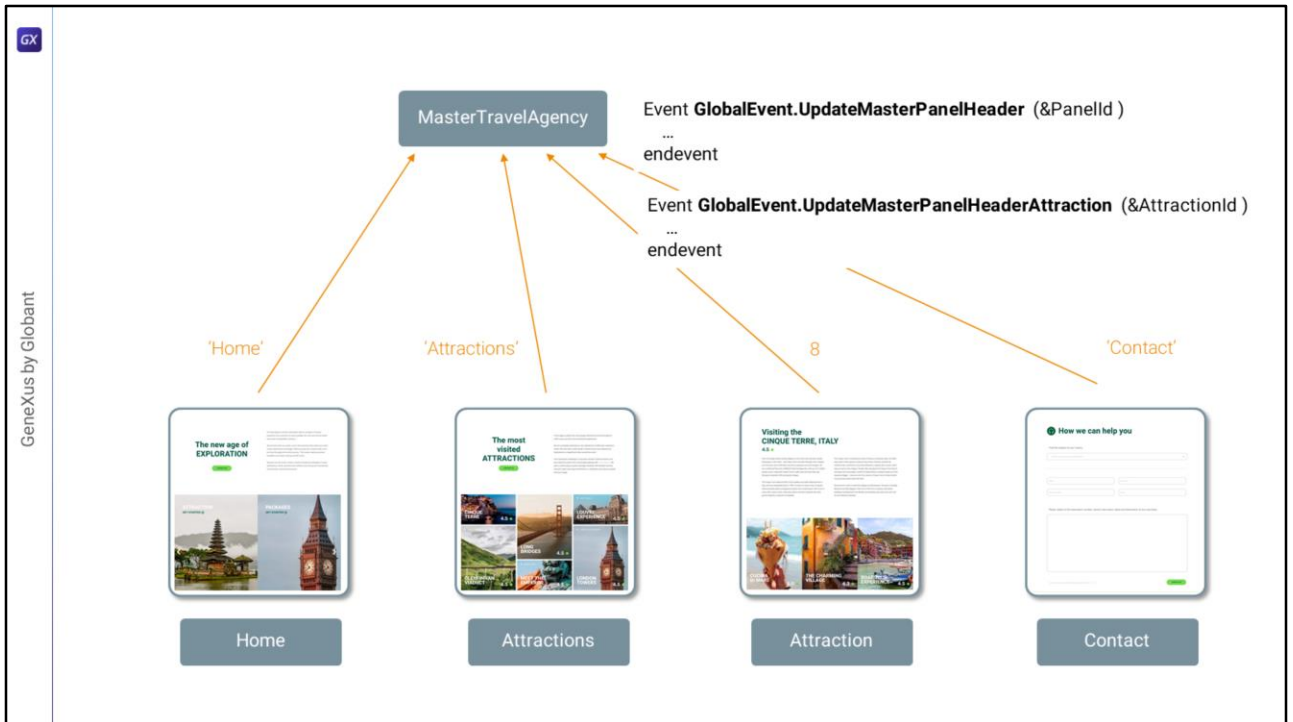
Aqui temos o Home, e se formos à opção de Attractions vemos aparecer o carrossel dos cards de atrações, com scroll horizontal...



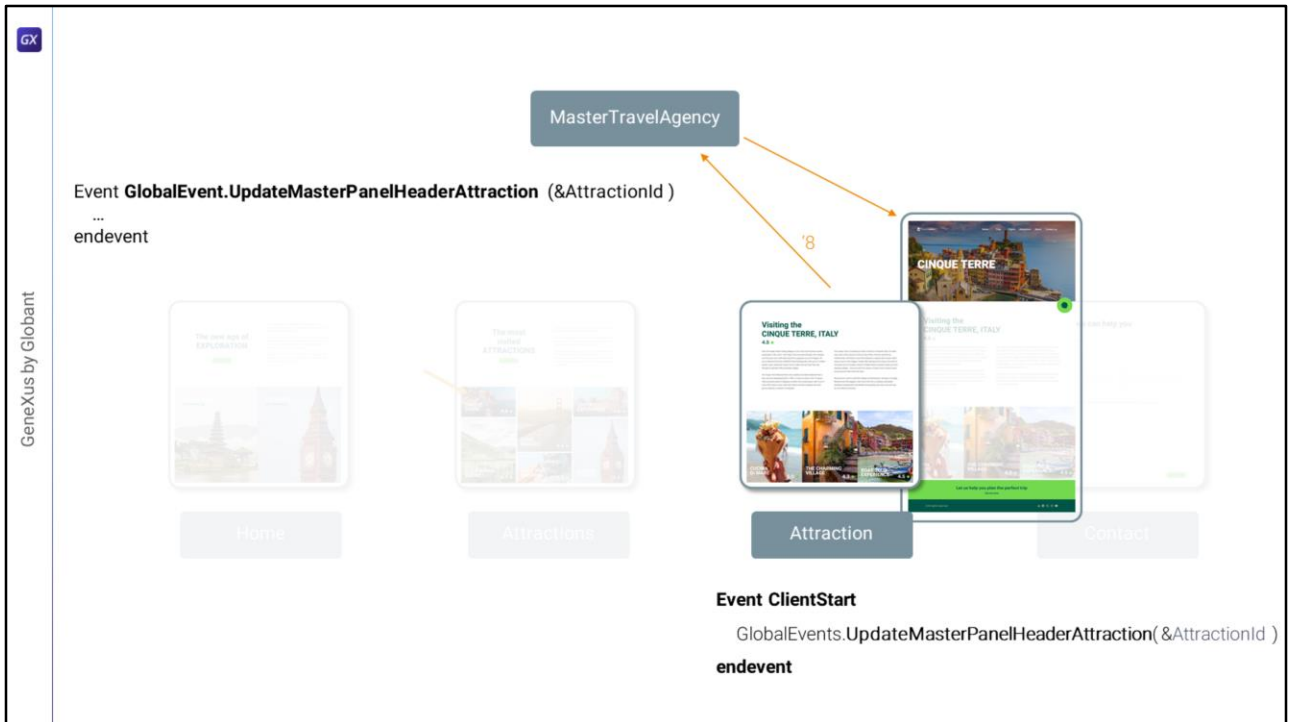
E se clicarmos em uma atração, ou tap, que é o mesmo... nos leva ao panel Attraction, que tem carregada a informação desta parte que é a única que implementamos até o momento.



Mas não vemos no Header do Master Panel nem a foto da atração nem o seu nome. Era nisso que queríamos nos concentrar neste vídeo, então vamos a isso.



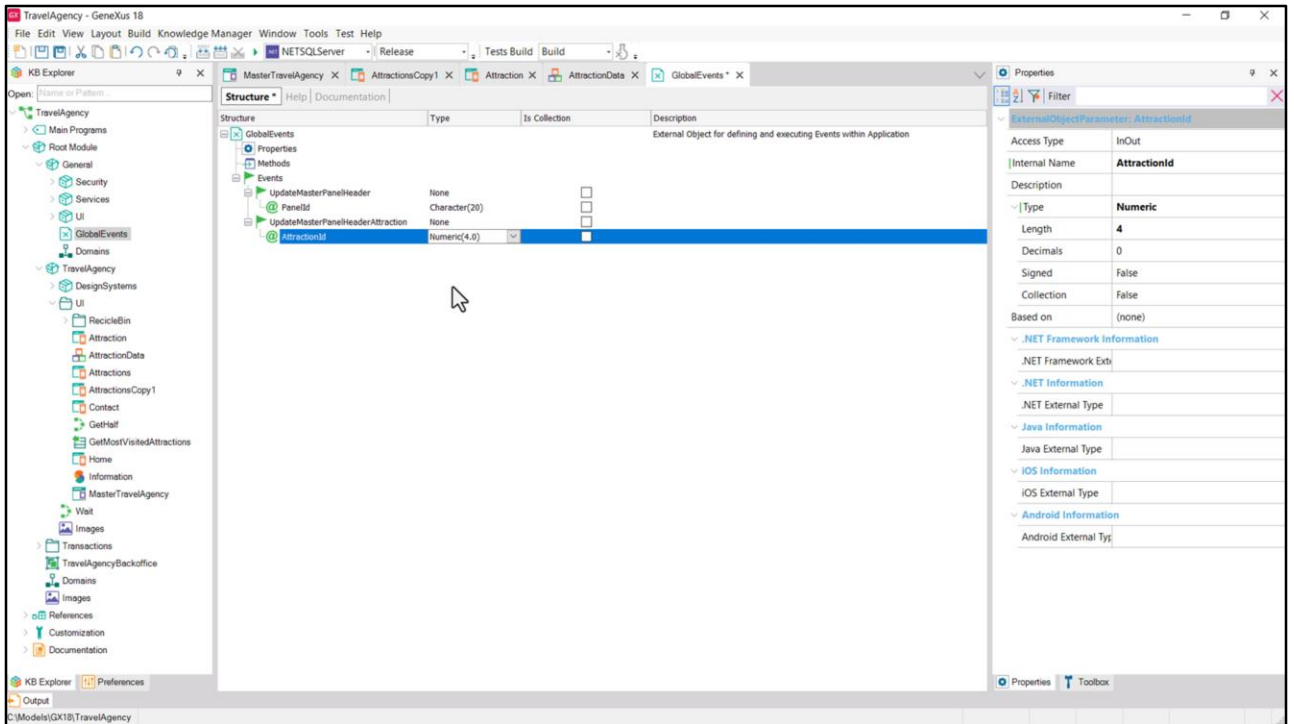
No vídeo anterior, havíamos decidido utilizar dois eventos globais, um para os painéis Home, Attractions e Contact, e um segundo evento global para Attraction.



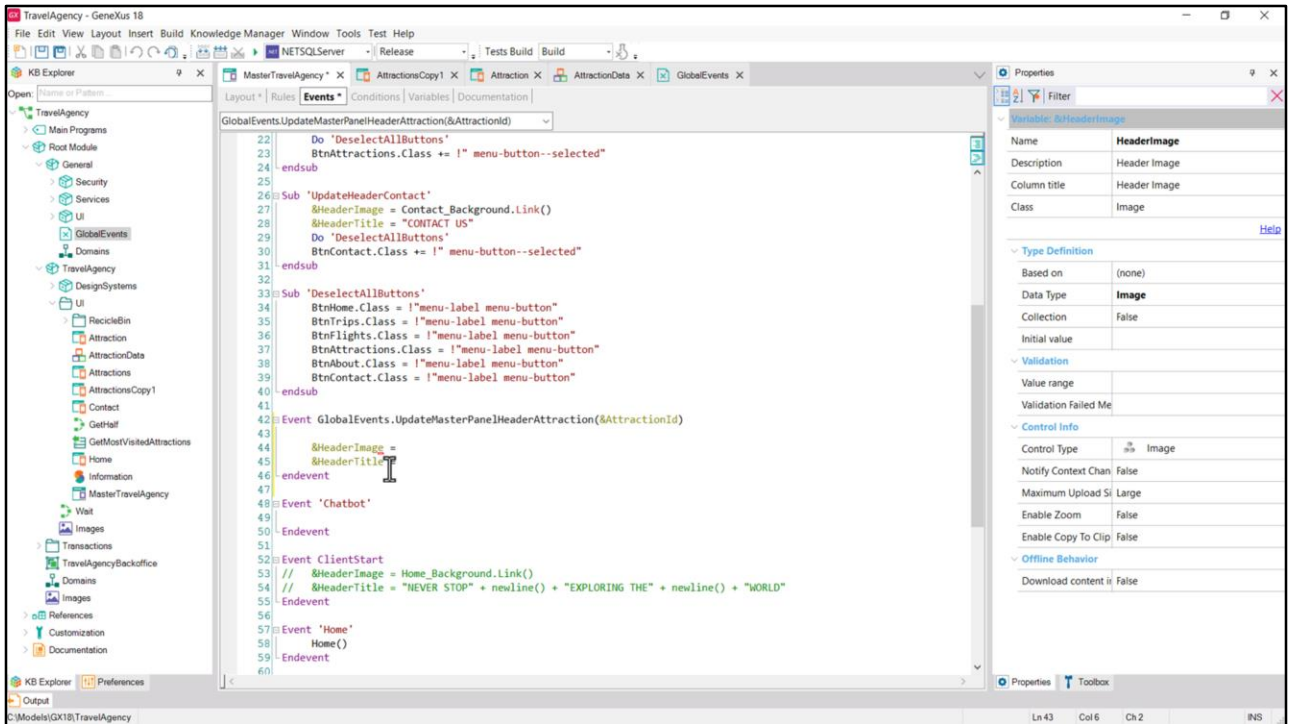
Depois o dispararemos do ClientStart de Attraction, passando o id da atração.

E teremos que implementá-lo no Master Panel, com sua lógica.

É o que faremos agora em GeneXus.



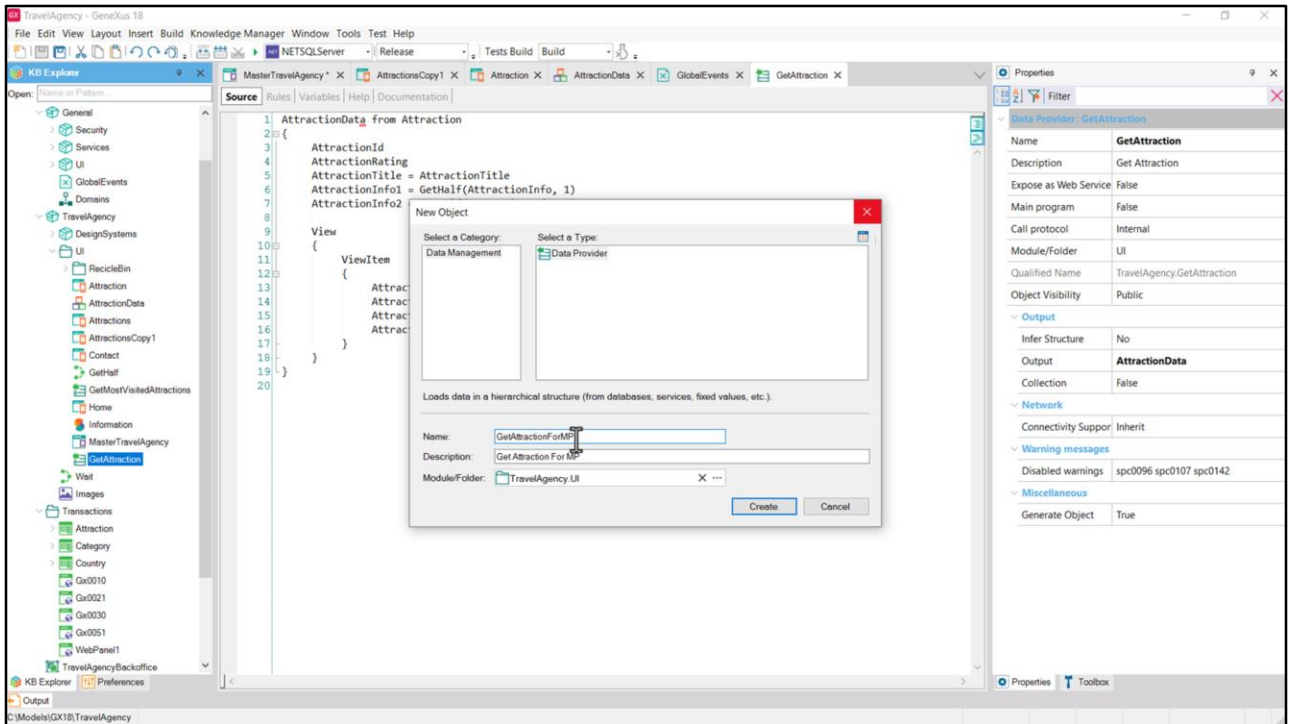
Abro o objeto GlobalEvents e adiciono este novo evento... com parâmetro AttractionId...



Agora vou ao Master Panel e o adiciono na seção de eventos, onde eu preferir.

Defino a variável `&AttractionId` neste objeto (lembramos que as variáveis são locais a cada objeto).

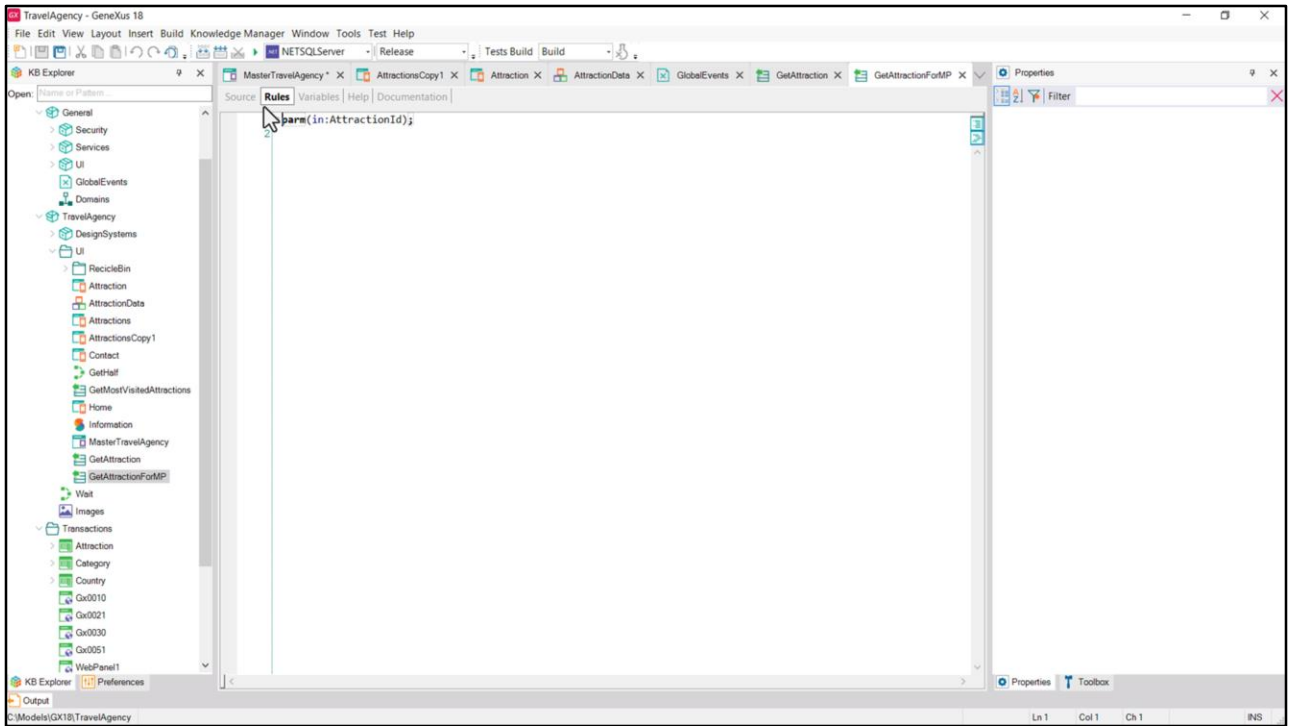
Bem. Preciso trazer da base de dados o nome da atração e sua foto, para atribuí-los a essas duas variáveis do Header do layout do Master Panel.



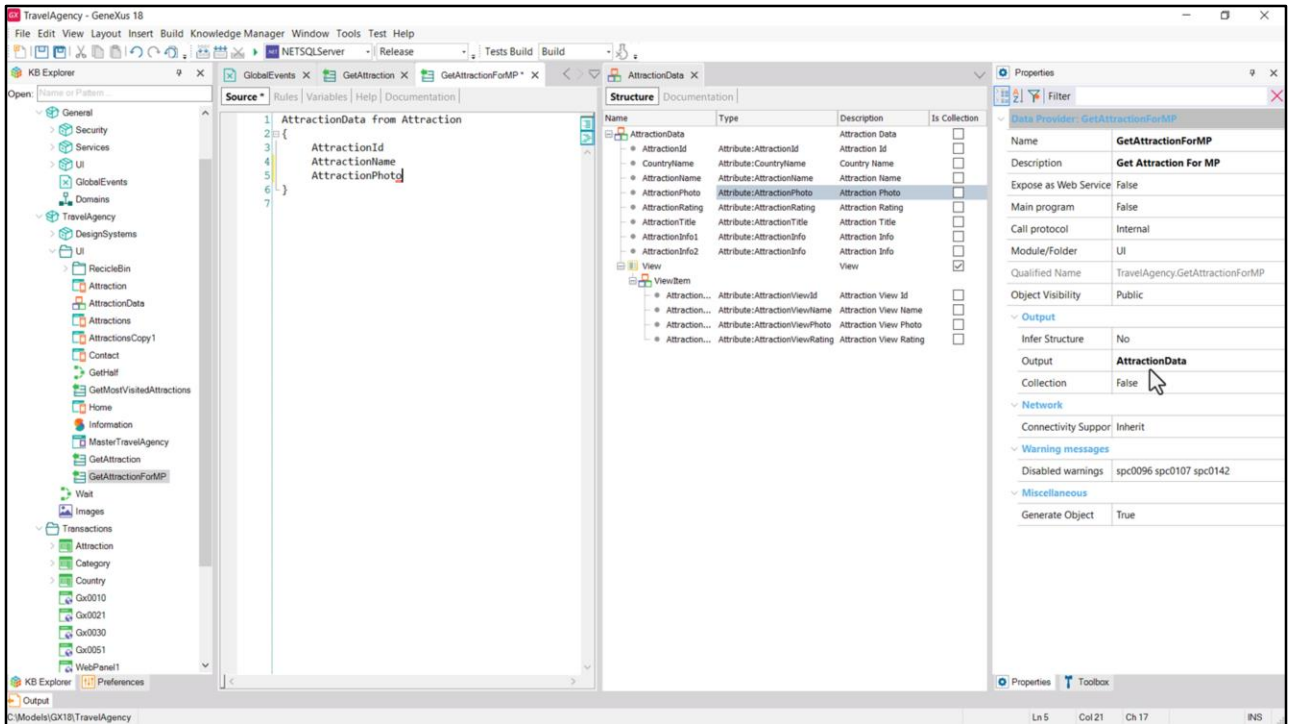
Se eu quiser ser extremamente eficiente, posso fazer um Save as do Data Provider que utilizei em Attraction para trazer a informação da atração, de modo a devolver carregados apenas esses dois valores... e todo o resto vazio.

Ficou erroneamente no folder Transaction, então vou movê-lo para este outro.

Faço o Save as, atribuo este nome...

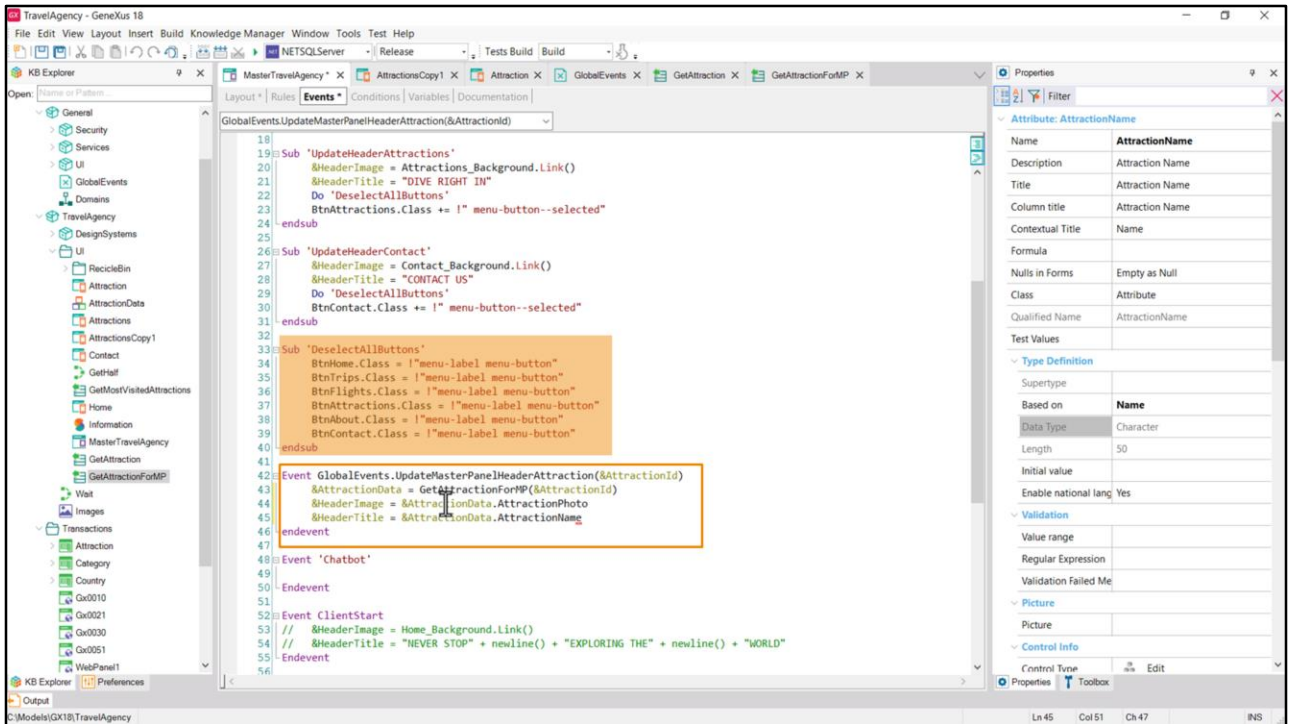


Receberá o parâmetro no atributo AttractionId...



...e no source removeremos todas essas atribuições e apenas deixaremos carregados os elementos que nos interessam... AttractionName e AttractionPhoto.

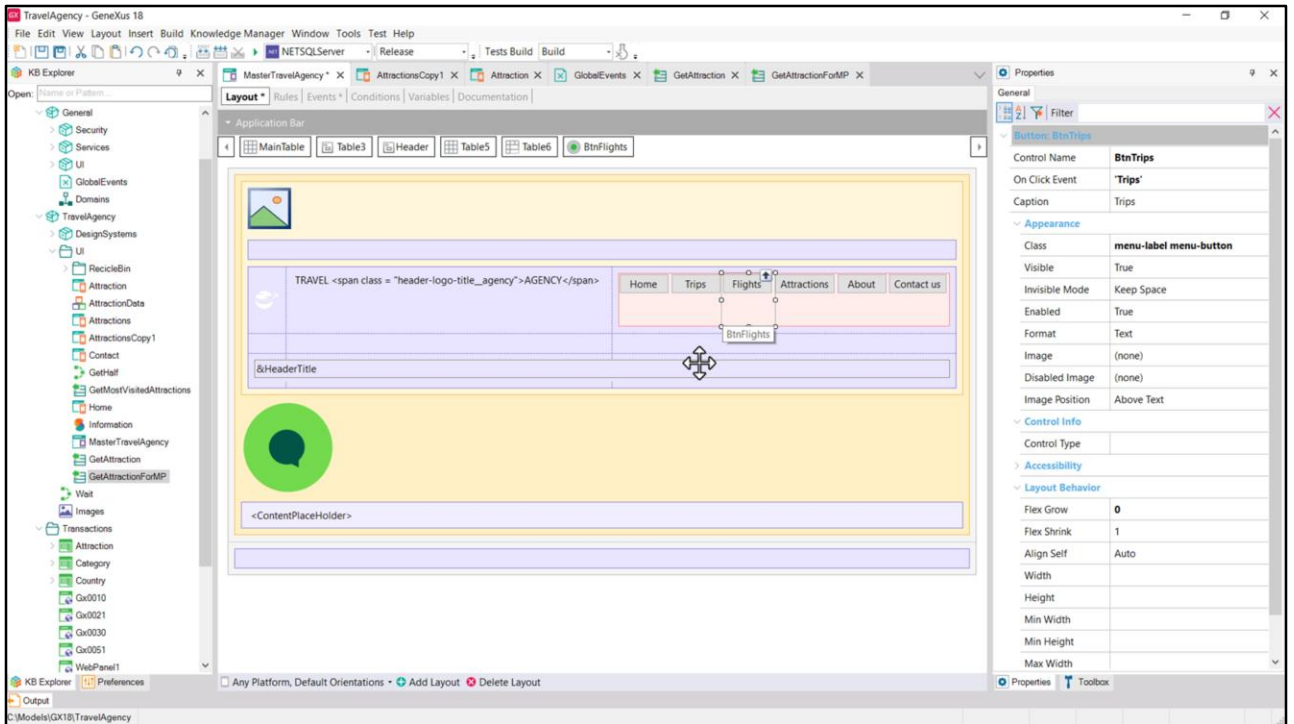
Devolverá este SDT carregado somente com esses três elementos. Todo o resto vazio.



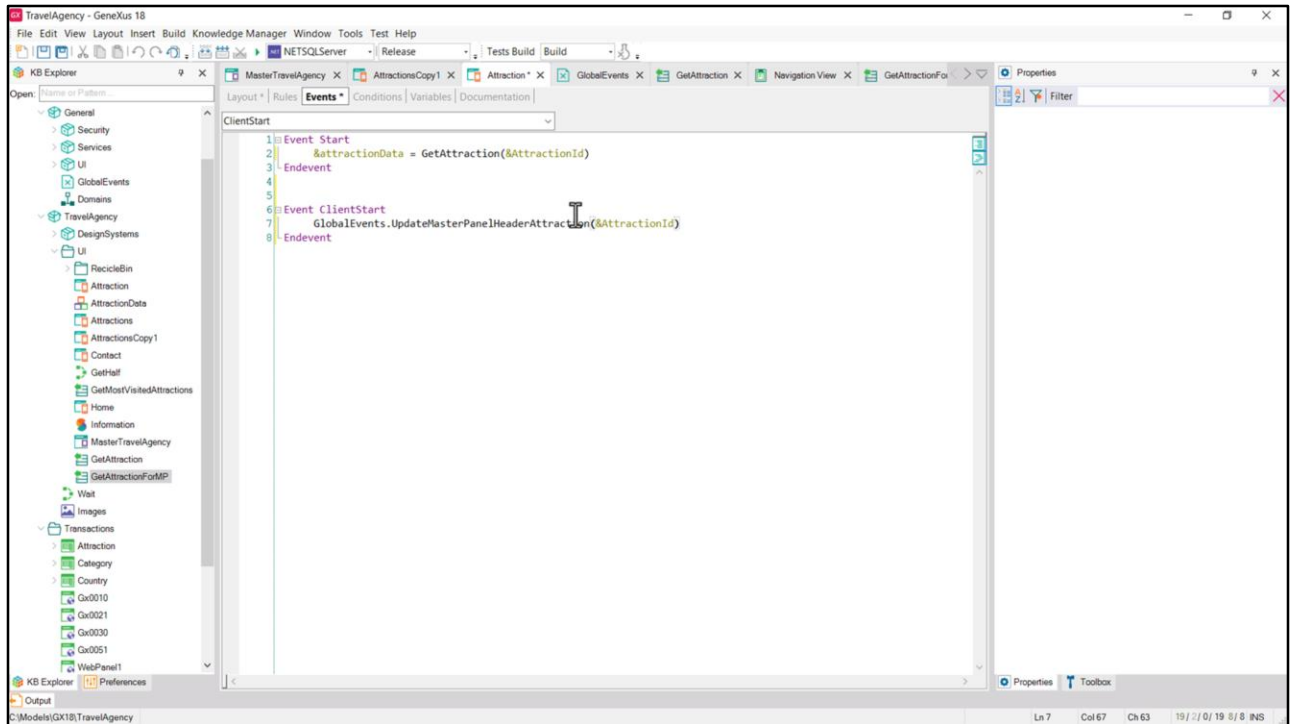
E depois, no Master Panel, a uma variável do tipo de dados do SDT, atribuímos o resultado de executar o data provider, passando como parâmetro o valor dessa variável &AttractionId, que é o parâmetro que recebe o evento global quando é disparado.

Depois, à variável &HeaderImage do layout, atribuímos do SDT, o elemento AttractionPhoto. E à &HeaderTitle, o elemento AttractionName.

E o que acontece com os botões do menu? Deveríamos chamar essa sub-rotina para desseleccioná-los?

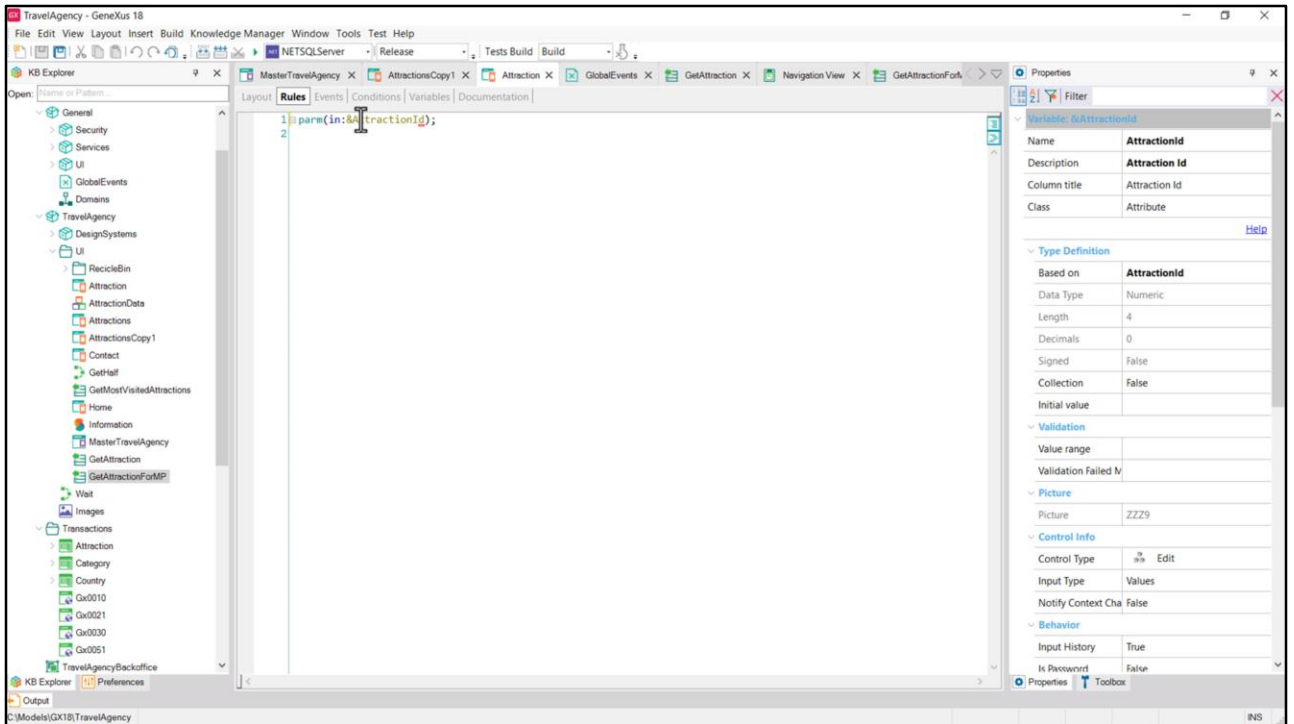


Na realidade, não é necessário porque essas mesmas classes já estão determinadas nos botões de forma estática.



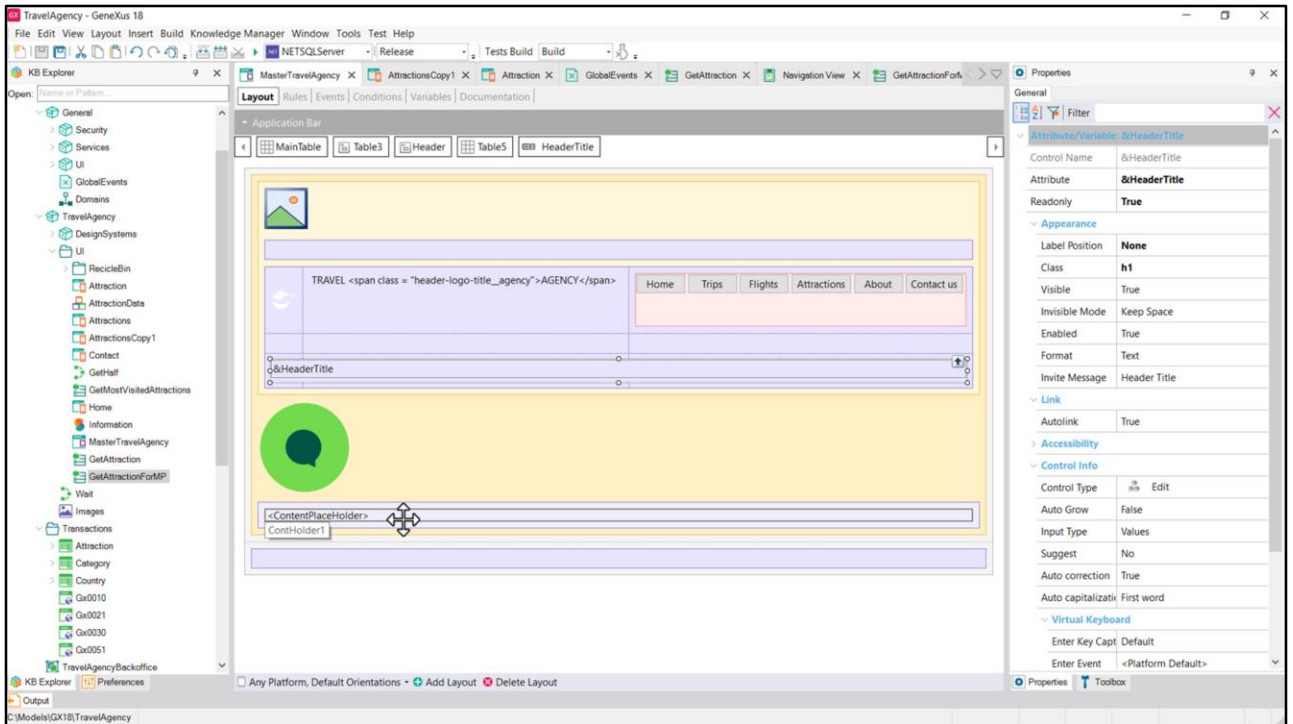
A única coisa que está faltando agora é disparar este evento a partir do panel Attraction.

Assim, no evento ClientStart... invocamos o evento global que queremos, passando como parâmetro esta variável, a que é recebida pelo panel Attraction. E a partir da qual, é carregado seu layout.



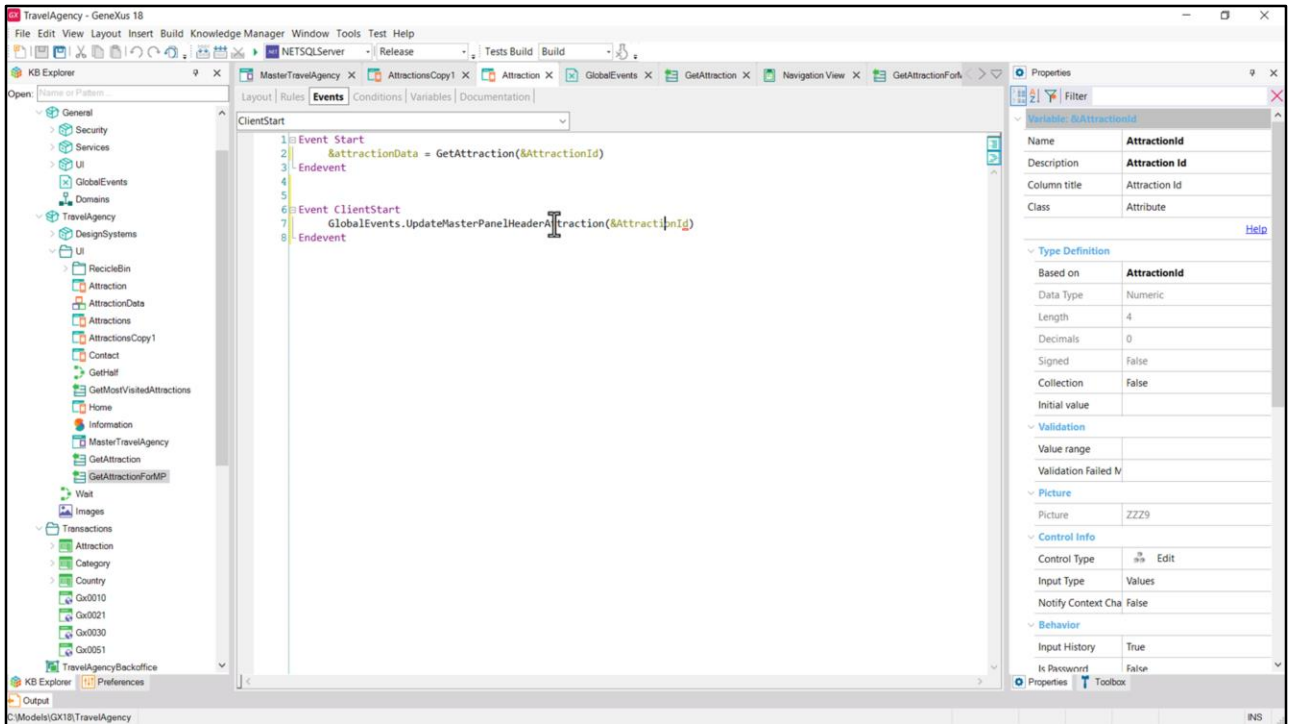
Em resumo, vamos revisar tudo.

Quando o panel Attraction é executado, recebe nesta variável o id da atração da qual se tratará.



A primeira coisa que faz, como sabíamos, é executar o Master Panel, e seu evento do sistema ClientStart, que neste caso está vazio. E renderiza pela primeira vez o Master Panel. Nesse momento, as variáveis do header ficarão vazias.

Em seguida, é executado para ser carregado no ContentPlaceHolder o panel Attraction propriamente dito.



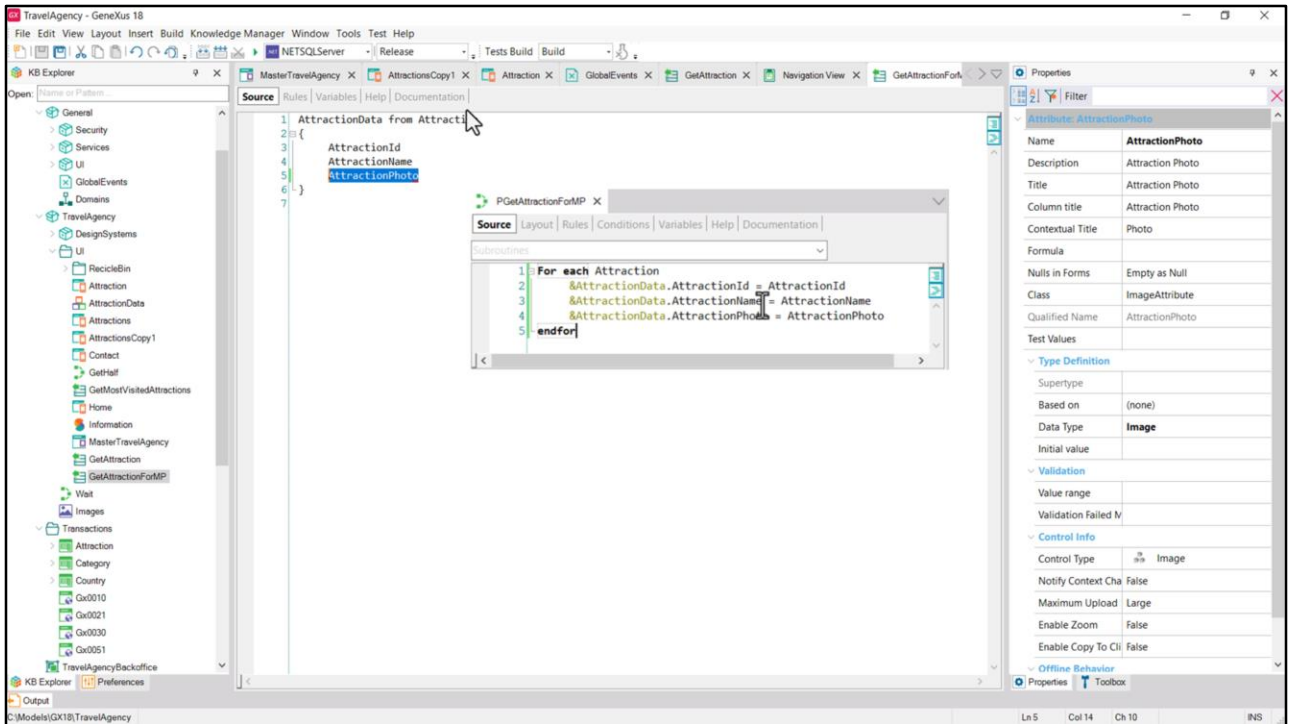
São atribuídas as propriedades estáticas dos controles do layout e, em seguida, é executado o evento ClientStart. É aqui que é disparado o evento global do Master Panel, que, como mencionamos no vídeo anterior, é assíncrono.

Já é enviado a este evento o id de atração recebido por parâmetro.


```
GlobalEvents.UpdateMasterPanelHeaderAttraction(&AttractionId)
19 Sub 'UpdateHeaderAttractions'
20   @HeaderImage = Attractions_Background.Link()
21   @HeaderTitle = "DIVE RIGHT IN"
22   Do 'DeselectAllButtons'
23     BtnAttractions.Class += "! menu-button--selected"
24   endsub
25
26 Sub 'UpdateHeaderContact'
27   @HeaderImage = Contact_Background.Link()
28   @HeaderTitle = "CONTACT US"
29   Do 'DeselectAllButtons'
30     BtnContact.Class += "! menu-button--selected"
31   endsub
32
33 Sub 'DeselectAllButtons'
34   BtnHome.Class = "!menu-label menu-button"
35   BtnTrips.Class = "!menu-label menu-button"
36   BtnFlights.Class = "!menu-label menu-label menu-button"
37   BtnAttractions.Class = "!menu-label menu-button"
38   BtnAbout.Class = "!menu-label menu-button"
39   BtnContact.Class = "!menu-label menu-button"
40 endsub
41
42 Event GlobalEvents.UpdateMasterPanelHeaderAttraction(&AttractionId)
43   @AttractionData = GetAttractionForMP(&AttractionId)
44   @HeaderImage = @AttractionData.AttractionPhoto
45   @HeaderTitle = @AttractionData.AttractionName
46 endevent
47
48 Event 'Chatbot'
49
50 Endevent
51
52 Event ClientStart
53 // @HeaderImage = Home_Background.Link()
54 // @HeaderTitle = "NEVER STOP" + newline() + "EXPLORING THE" + newline() + "WORLD"
55 Endevent
56
57 Event 'Home'
```

E aqui é que executa no Master Panel, isso. Aqui é onde se carrega a imagem e o título a partir do SDT que devolveu carregado este data provider.

Já aviso que acabou de ser corrigido um bug que fazia com que um DataProvider que carregava uma imagem, não o fizesse corretamente. Se ao executar isso acontecer de não verem em execução a imagem, então é porque possuem uma versão com o bug.



Nesse caso, recomendo utilizar um procedimento para carregar o SDT em vez de um Data Provider, e assim não terão nenhum problema.

The screenshot shows the Genexus 18 IDE with the following components:

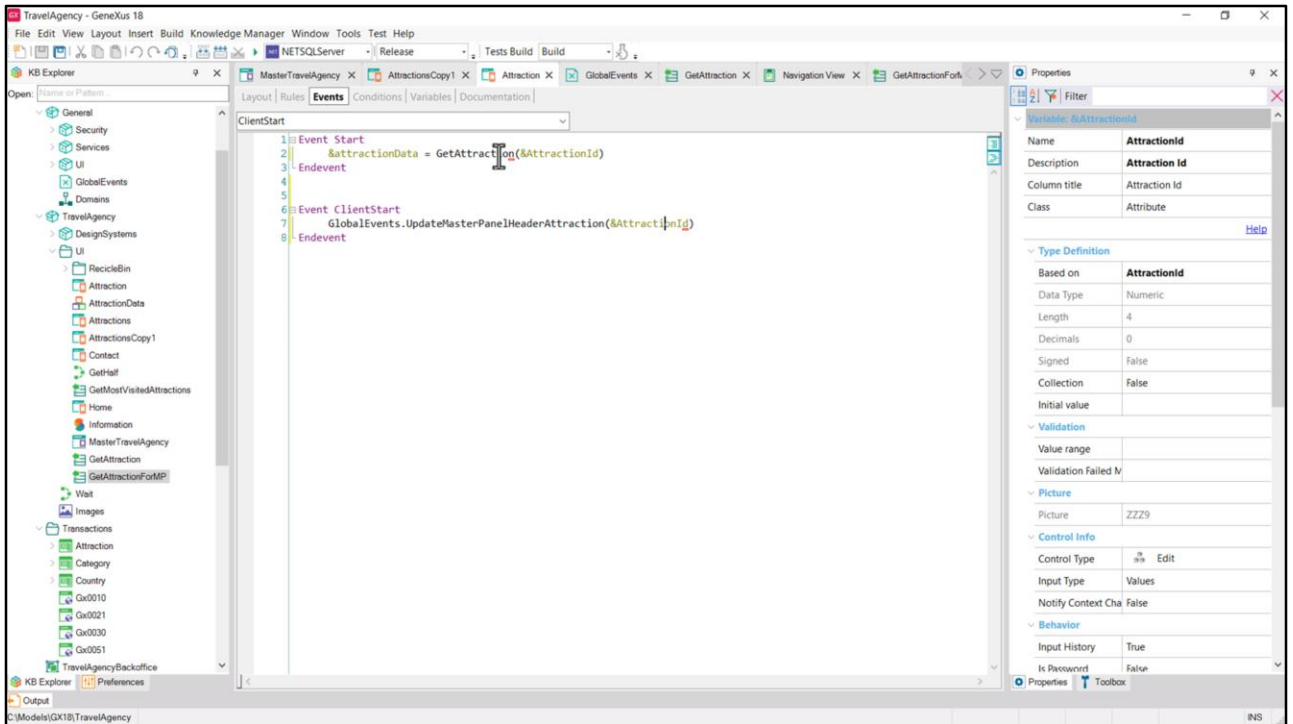
- KB Explorer:** A tree view on the left showing the project structure, including folders like 'General', 'Security', 'Services', 'UI', 'GlobalEvents', 'Domains', 'TravelAgency', 'DesignSystems', 'UI', 'RecycleBin', 'Attraction', 'AttractionData', 'Attractions', 'AttractionsCopy1', 'Contact', 'GetHalf', 'GetMostVisitedAttractions', 'Home', 'Information', 'MasterTravelAgency', 'GetAttraction', 'GetAttractionForMP', 'Wait', 'Images', 'Transactions', 'Attraction', 'Category', 'Country', 'Gx0010', 'Gx0021', 'Gx0030', 'Gx0051', 'TravelAgencyBackoffice', and 'TravelAgencyBackoffice'.
- Code Editor:** The main window displays the 'Events' tab for the 'GlobalEvents.UpdateMasterPanelHeaderAttraction(&AttractionId)' event. The code includes several subroutines and an event handler:

```
19 Sub 'UpdateHeaderAttractions'
20   &HeaderImage = Attractions_Background.Link()
21   &HeaderTitle = "DIVE RIGHT IN"
22   Do 'DeselectAllButtons'
23   BtnAttractions.Class += "! menu-button--selected"
24 Endsub
25
26 Sub 'UpdateHeaderContact'
27   &HeaderImage = Contact_Background.Link()
28   &HeaderTitle = "CONTACT US"
29   Do 'DeselectAllButtons'
30   BtnContact.Class += "! menu-button--selected"
31 Endsub
32
33 Sub 'DeselectAllButtons'
34   BtnHome.Class = "!menu-label menu-button"
35   BtnTrips.Class = "!menu-label menu-button"
36   BtnFlights.Class = "!menu-label menu-button"
37   BtnAttractions.Class = "!menu-label menu-button"
38   BtnAbout.Class = "!menu-label menu-button"
39   BtnContact.Class = "!menu-label menu-button"
40 Endsub
41
42 Event GlobalEvents.UpdateMasterPanelHeaderAttraction(&AttractionId)
43   &AttractionData = GetAttractionForMP(&AttractionId)
44   &HeaderImage = &AttractionData.AttractionPhoto
45   &HeaderTitle = &AttractionData.AttractionName
46 Endevent
47
48 Event 'Chatbot'
49 Endevent
50
51
52 Event ClientStart
53 // &HeaderImage = Home_Background.Link()
54 // &HeaderTitle = "NEVER STOP" + newline() + "EXPLORING THE" + newline() + "WORLD"
55 Endevent
56
57 Event 'Home'
```
- Properties Window:** On the right, the 'Properties' window is open for the variable '&AttractionData'. It shows the following details:

Name	AttractionData
Description	Attraction Data
Column title	Attraction Data
Class	Attribute

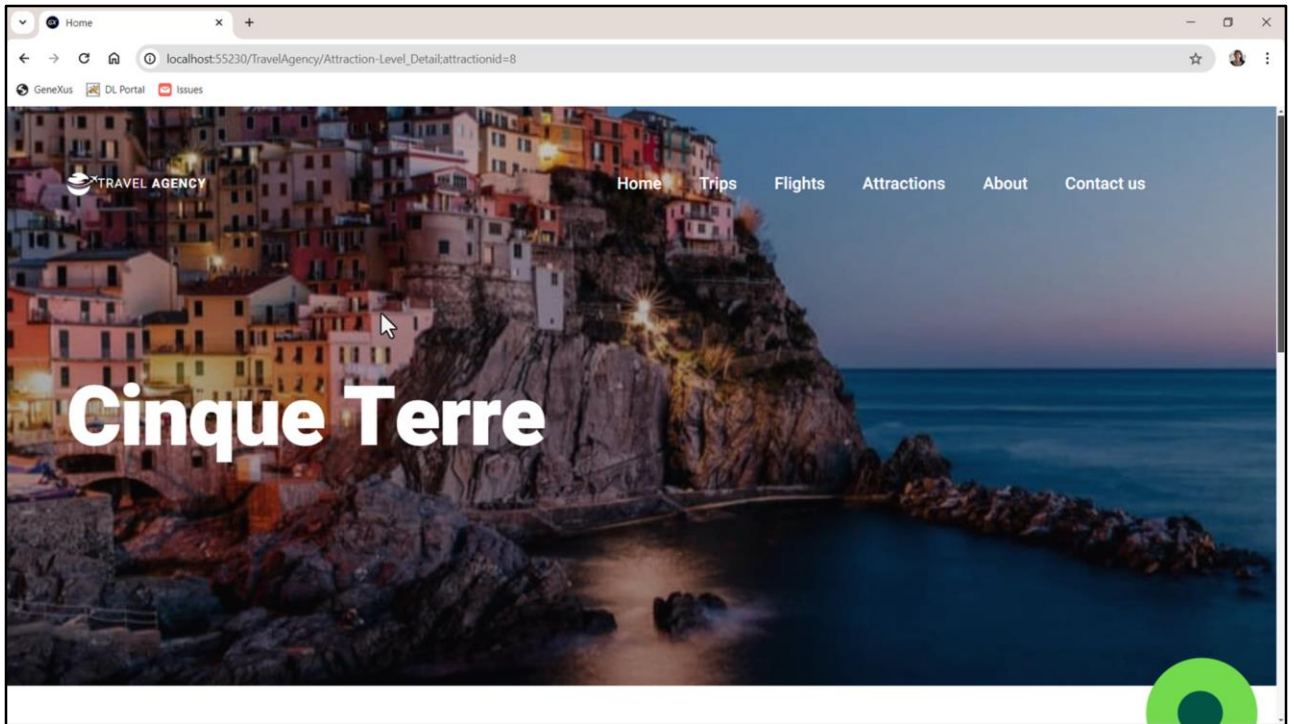
Below the table, there are sections for 'Type Definition', 'Validation', 'Control Info', 'Behavior', and 'Appearance'.

Bom, então, aqui será renderizada a imagem e o título do Header do Master Panel.

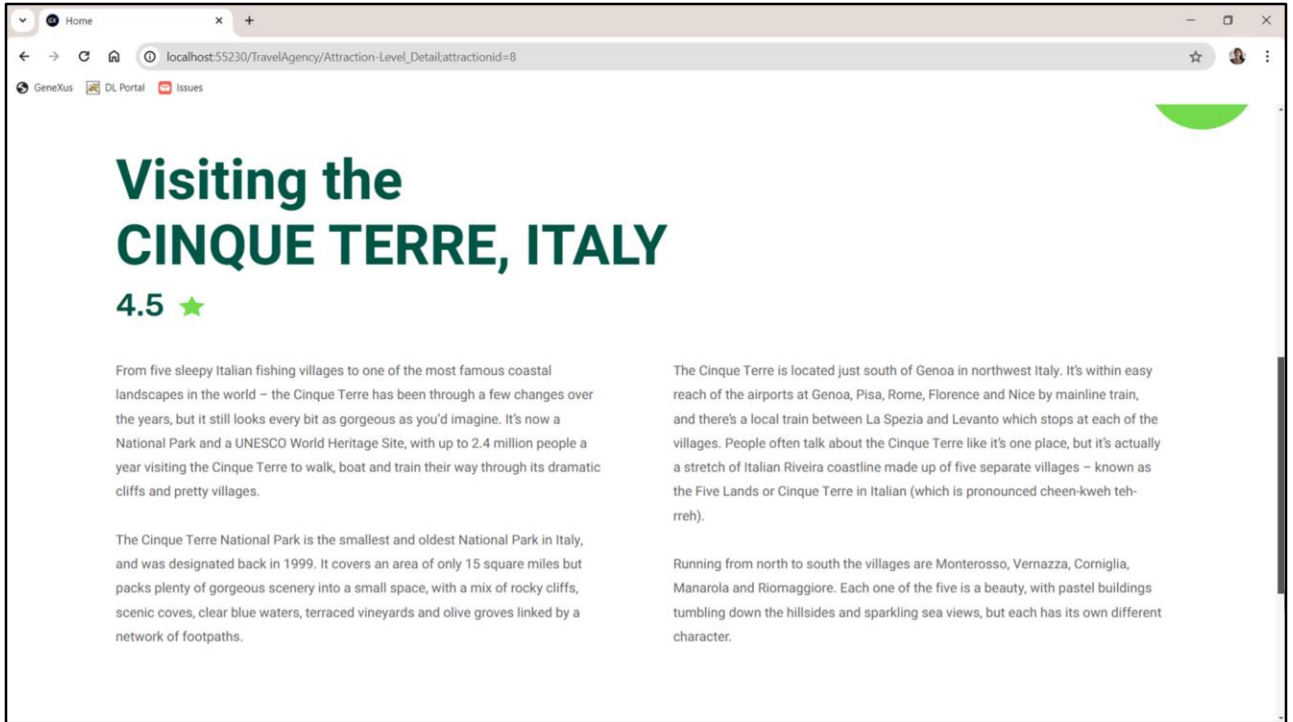


E por outro lado, em Attraction já terá continuado com o evento do sistema seguinte, o Start. Ali, chama-se este outro Data Provider que traz carregada a informação necessária da atração deste id. E essa é a que será renderizada no layout. Lembremos que os eventos Refresh e Load estão vazios neste panel.

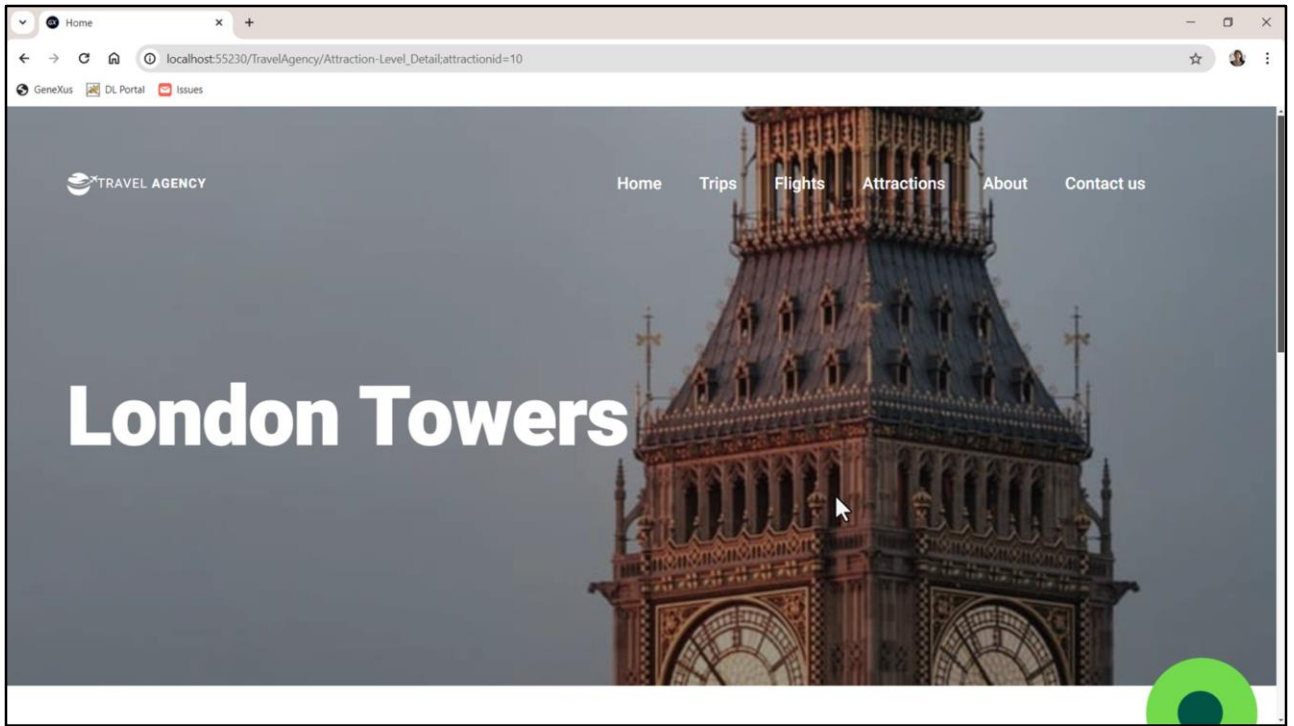
Bem, executamos.



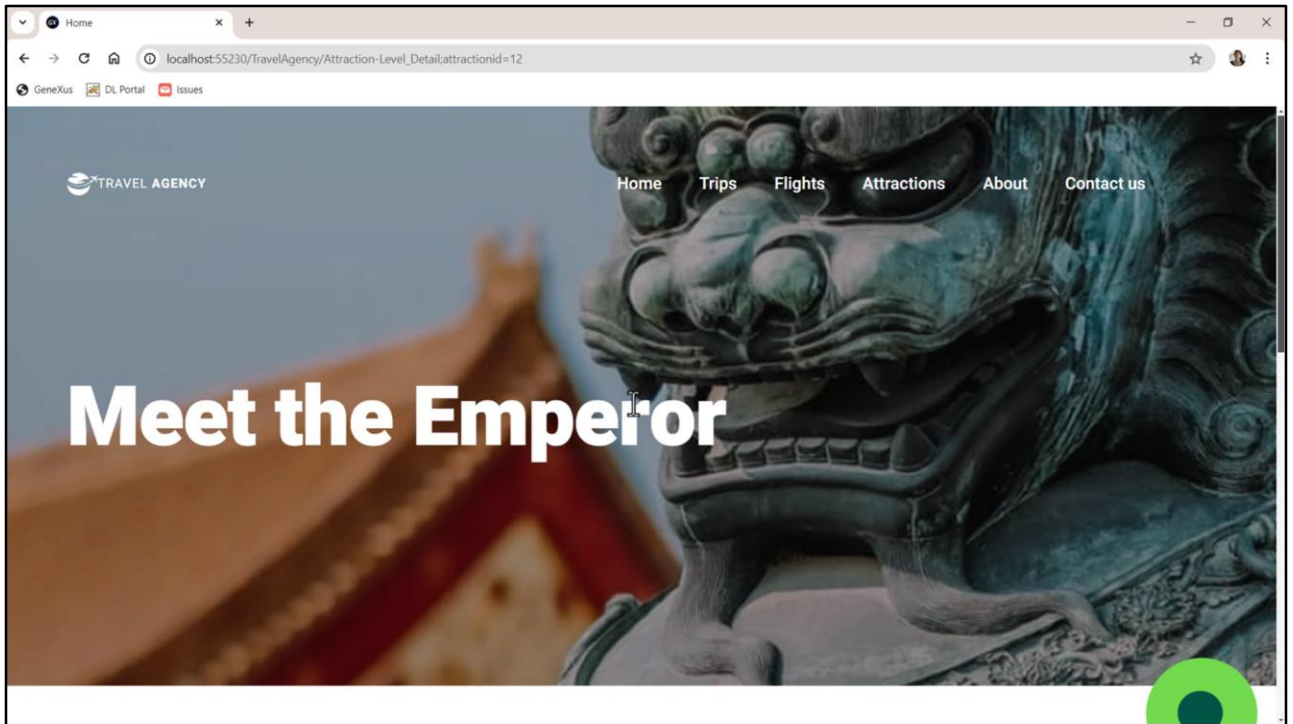
Vamos a Attractions... escolhemos a primeira do carrossel... e vemos como são carregados corretamente foto e título...



...e o conteúdo específico da página da atração.

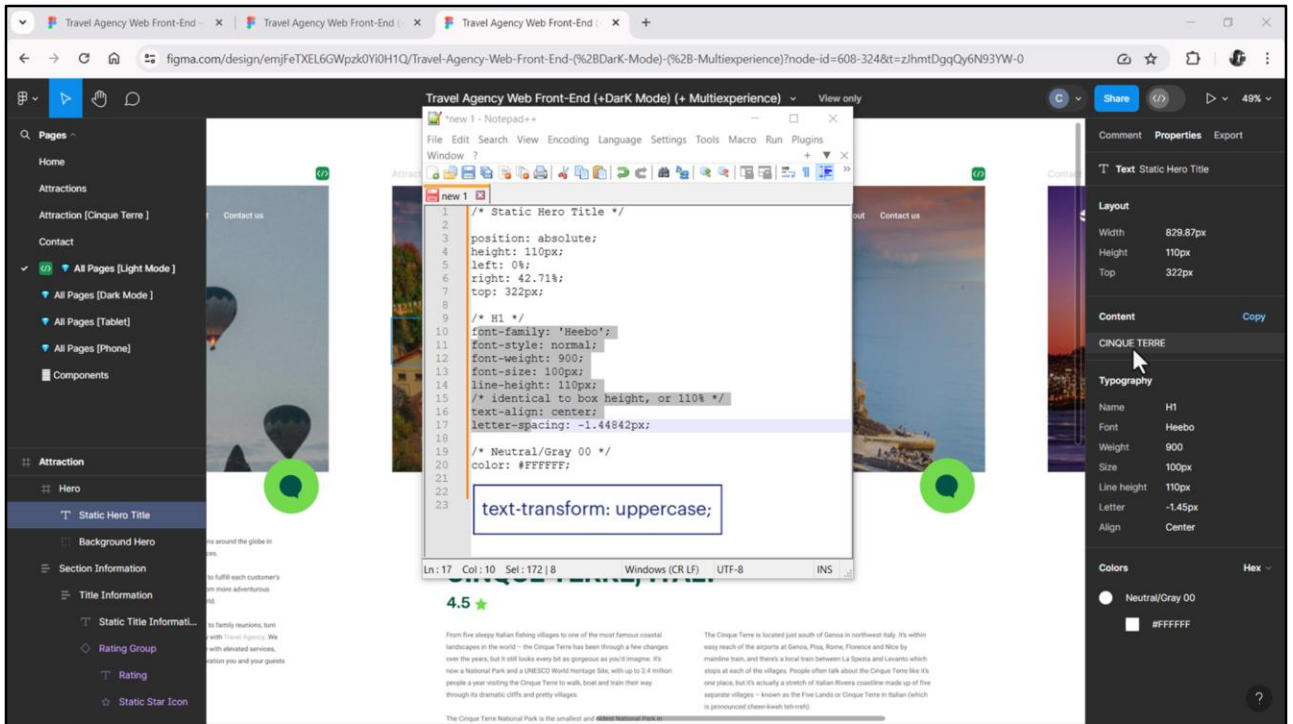


Se agora vamos escolher outra... também...

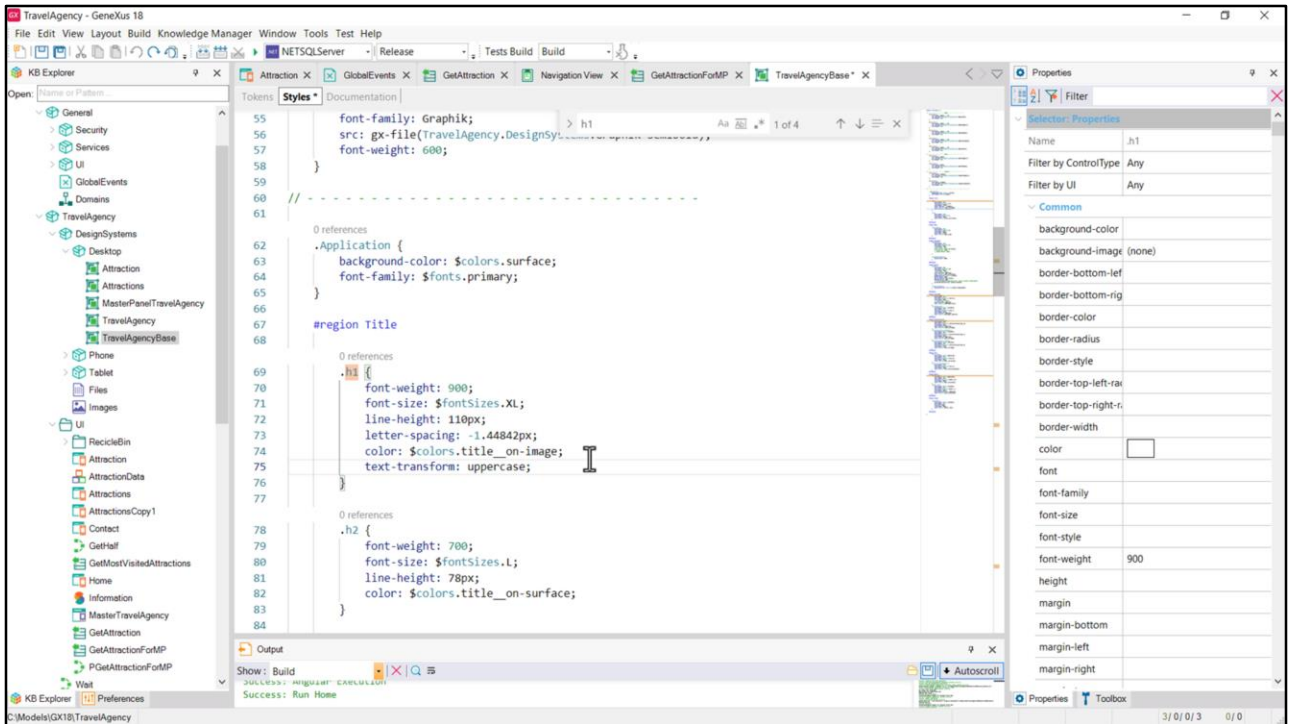


Escolhemos outra... e também....

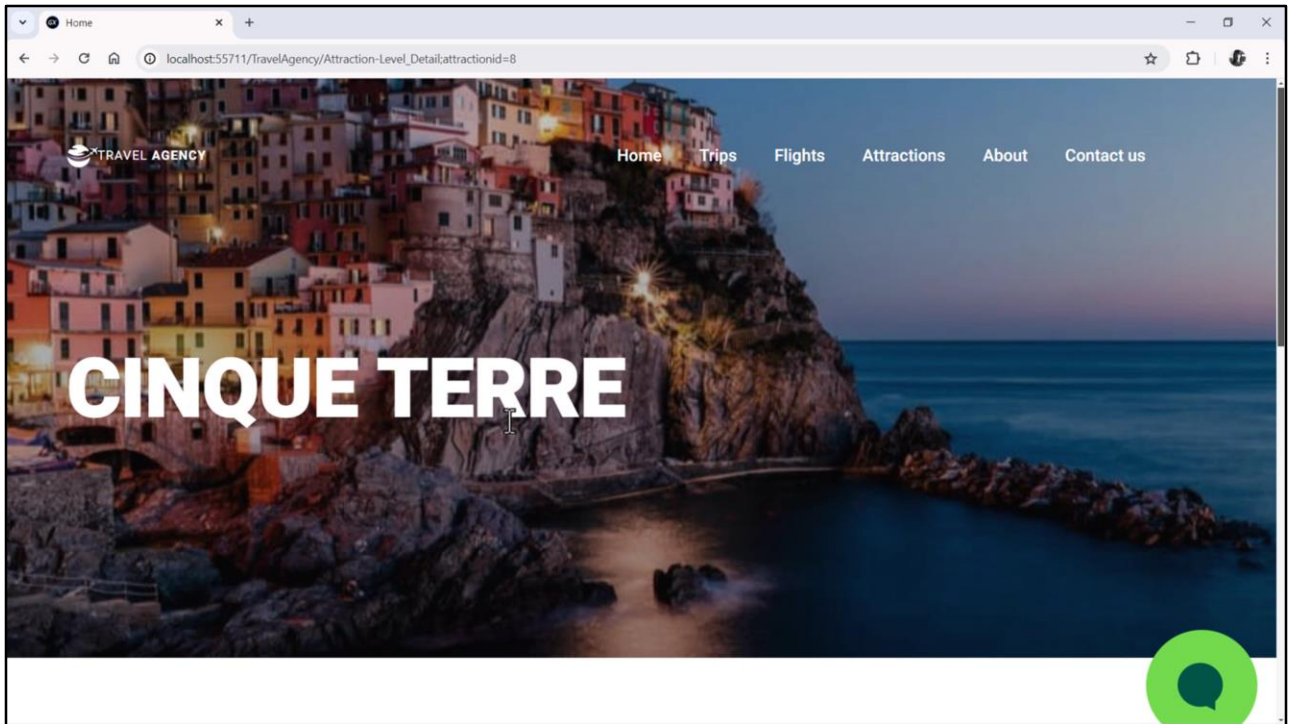
Porém, este título deveria ser exibido sempre em maiúsculas, independentemente de como esteja inserido na variável.



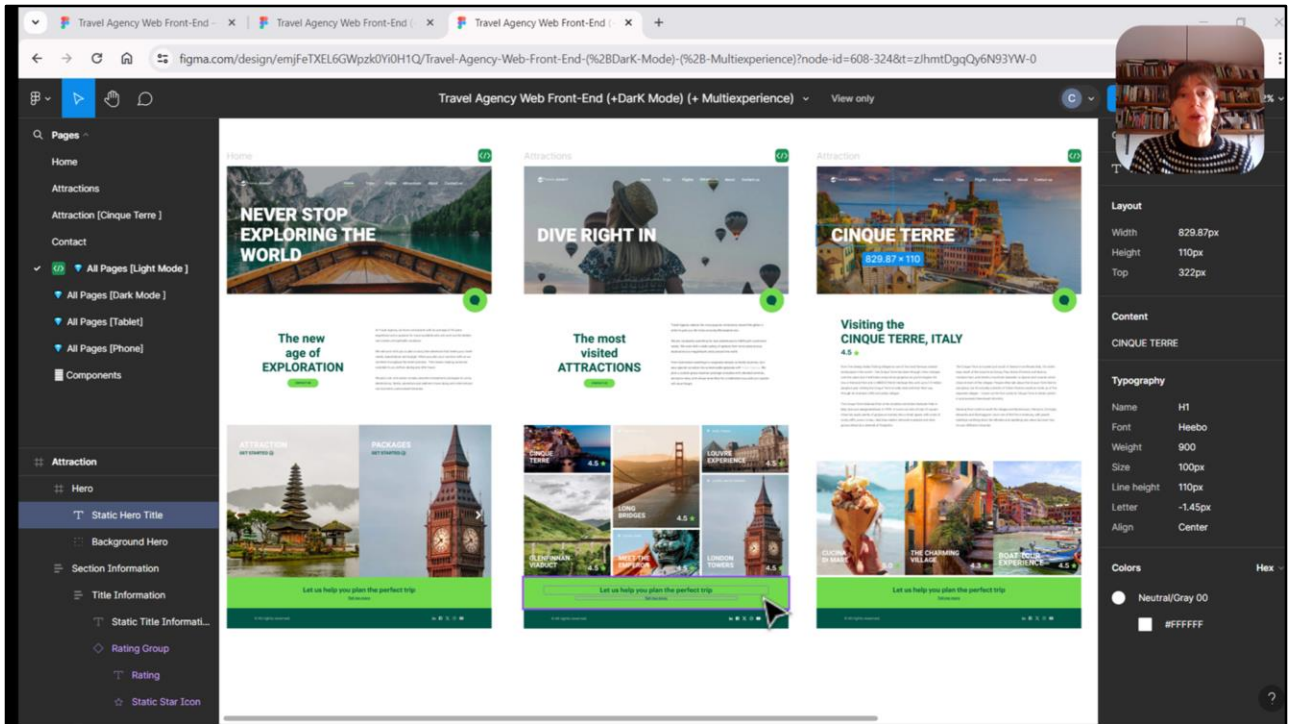
Evidentemente Chechu não percebeu quando criou o estilo tipográfico... porque inseriu os textos em maiúsculas... então não adicionou a propriedade text-transform: uppercase.



Vamos adicioná-la à classe h1.



Bom, agora sim. Certo.



Bom, e com isso encerro a implementação do Master Panel.

Fica pendente, e deixo como tarefa para vocês, implementar o footer. Toda essa parte de baixo, que é muito simples com todo o conhecimento que temos até o momento.

No vídeo seguinte, vamos nos dedicar à implementação, agora sim em detalhe, deste carrossel. É um carrossel que utiliza, como vimos, a informação da base de dados, portanto é o mais complexo dos três que existem. Por isso vou me concentrar neste. Os demais serão bastante simples a partir do momento em que entendermos como implementar e desenhar este carrossel.

Bom, espero vocês no próximo vídeo.

GX

GeneXus by Globant

GeneXus[™]
by Globant

training.genexus.com